IBM Security QRadar

Log Sources User Guide

Version 7.2.5

IBM Security QRadar

Log Sources User Guide

Version 7.2.5

Note

Before using this information and the product that it supports, read the information in “Notices” on page 55.

Product information

This document applies to IBM QRadar Security Intelligence Platform V7.2.5 and subsequent releases unless superseded by an updated version of this document.

Contents

About this guide

Log sources are third-party devices that send events to IBM® Security QRadar® for collection, storage, parsing, and processing.

Intended audience

Administrators must have QRadar access and knowledge of the corporate network and networking technologies.

Technical documentation

To find IBM Security QRadar product documentation on the web, including all translated documentation, access the IBM Knowledge Center (http://www.ibm.com/support/knowledgecenter/SS42VS/welcome).

For information about how to access more technical documentation in the QRadar products library, see Accessing IBM Security Documentation Technical Note (www.ibm.com/support/docview.wss?rs=0&uid=swg21614644).

Contacting customer support

For information about contacting customer support, see the Support and Download Technical Note (http://www.ibm.com/support/docview.wss?rs=0&uid=swg21612861).

Statement of good security practices

IT system security involves protecting systems and information through prevention, detection and response to improper access from within and outside your enterprise. Improper access can result in information being altered, destroyed, misappropriated or misused or can result in damage to or misuse of your systems, including for use in attacks on others. No IT system or product should be considered completely secure and no single product, service or security measure can be completely effective in preventing improper use or access. IBM systems, products and services are designed to be part of a comprehensive security approach, which will necessarily involve additional operational procedures, and may require other systems, products or services to be most effective. IBM DOES NOT WARRANT THAT ANY SYSTEMS, PRODUCTS OR SERVICES ARE IMMUNE FROM, OR WILL MAKE YOUR ENTERPRISE IMMUNE FROM, THE MALICIOUS OR ILLEGAL CONDUCT OF ANY PARTY.

Chapter 1. Introduction to log source management

You can configure IBM Security QRadar to accept event logs from log sources that are on your network. A *log source* is a data source that creates an event log.

For example, a firewall or intrusion protection system (IPS) logs security-based events, and switches or routers logs network-based events.

To receive raw events from log sources, QRadar supports many protocols. *Passive protocols* listen for events on specific ports. *Active protocols* use APIs or other communication methods to connect to external systems that poll and retrieve events.

Depending on your license limits, QRadar can read and interpret events from more than 300 log sources.

To configure a log source for QRadar, you must do the following tasks:

1. Download and install a device support module (DSM) that supports the log source. A *DSM* is software application that contains the event patterns that are required to identify and parse events from the original format of the event log to the format that QRadar can use. For more information about DSMs and the supported log sources, see the *DSM Configuration Guide.*
2. If automatic discovery is supported for the DSM, wait for QRadar to automatically add the log source to your list of configured log sources.
3. If automatic discover is not supported for the DSM, manually create the log source configuration.

Adding a log source

If a log source is not automatically discovered, you can manually add a log source to receive events from your network devices or appliances.

About this task

The following table describes the common log source parameters for all log source types:

Table 1. Log source parameters

Parameter	Description
Log Source Identifier	The IPv4 address or host name that identifies the log source. If your network contains multiple devices that are attached to a single management console, specify the IP address of the individual device that created the event. A unique identifier for each, such as an IP address, prevents event searches from identifying the management console as the source for all of the events.

Table 1. Log source parameters (continued)

Parameter	Description
Enabled	When this option is not enabled, the log source does not collect events and the log source is not counted in the license limit.
Credibility	Credibility is a representation of the integrity or validity of events that are created by a log source. The credibility value that is assigned to a log source can increase or decrease based on incoming events or adjusted as a response to user-created event rules. The credibility of events from log sources contributes to the calculation of the offense magnitude and can increase or decrease the magnitude value of an offense.
Target Event Collector	Specifies the QRadar Event Collector that polls the remote log source. Use this parameter in a distributed deployment to improve Console system performance by moving the polling task to an Event Collector.
Coalescing Events	Increases the event count when the same event occurs multiple times within a short time interval. Coalesced events provide a way to view and determine the frequency with which a single event type occurs on the **Log Activity** tab. When this check box is clear, events are viewed individually and events are not bundled. New and automatically discovered log sources inherit the value of this check box from the **System Settings** configuration on the **Admin** tab. You can use this check box to override the default behavior of the system settings for an individual log source.

Procedure

1. Click the **Admin** tab.
2. Click the **Log Sources** icon.
3. Click **Add**.
4. Configure the common parameters for your log source.
5. Configure the protocol-specific parameters for your log source.
6. Click **Save**.
7. On the **Admin** tab, click **Deploy Changes**.

JDBC protocol configuration options

QRadar uses the JDBC protocol to collect information from tables or views that contain event data from several database types.

The following table describes the protocol-specific parameters for the JDBC protocol:

Table 2. JDBC protocol parameters

Parameter	Description
Database Type	From the list box, select the type of database that contains the events.
Database Name	The database name must match the database name that is specified in the **Log Source Identifier** field.
Port	The JDBC port must match the listen port that is configured on the remote database. The database must permit incoming TCP connections. If a **Database Instance** is used with the MSDE database type, administrators must leave the **Port** parameter blank in the log source configuration.
Username	A user account for QRadar in the database.
Authentication Domain	A domain must be configured for MSDE databases that are within a Windows domain. If your network does not use a domain, leave this field blank.
Database Instance	The database instance, if required. MSDE databases can include multiple SQL server instances on one server. When a non-standard port is used for the database or access is blocked to port 1434 for SQL database resolution, the **Database Instance** parameter must be blank in the log source configuration.
Predefined Query	Optional.
Table Name	The name of the table or view that includes the event records. The table name can include the following special characters: dollar sign ($), number sign (#), underscore (_), en dash (-), and period (.).
Select List	The list of fields to include when the table is polled for events. You can use a comma-separated list or type * to select all fields from the table or view. If a comma-separated list is defined, the list must contain the field that is defined in the **Compare Field**.
Compare Field	A numeric value or time stamp field from the table or view that identifies new events that are added to the table between queries. Enables the protocol to identify events that were previously polled by the protocol to ensure that duplicate events are not created.
Use Prepared Statements	Prepared statements enable the JDBC protocol source to set up the SQL statement, and then run the SQL statement numerous times with different parameters. For security and performance reasons, most JDBC protocol configurations can use prepared statements.
Start Date and Time	If a start time is not defined, the protocol attempts to poll for events after the log source configuration is saved and deployed.
Polling Interval	The default polling interval is 10 seconds.
EPS Throttle	The upper limit for the permitted number of Events Per Second (EPS).
Database Locale	For multilingual installations, use the **Database Locale** field to specify the language to use.

Table 2. JDBC protocol parameters (continued)

Parameter	Description
Database Codeset	For multilingual installations, use the **Codeset** field to specify the character set to use.
Use Named Pipe Communication	Named pipe connections for MSDE databases require that the user name and password field use a Windows authentication user name and password instead of the database user name and password. The log source configuration must use the default named pipe on the MSDE database.
Use NTLMv2	The **Use NTLMv2** check box does not interrupt communications for MSDE connections that do not require NTLMv2 authentication.

JDBC SiteProtector configuration options

You can configure log sources to use the Java™ Database Connectivity (JDBC) SiteProtector™ protocol to remotely poll IBM Proventia® Management SiteProtector® databases for events.

The JDBC - SiteProtector protocol combines information from the SensorData1 and SensorDataAVP1 tables in the creation of the log source payload. The SensorData1 and SensorDataAVP1 tables are in the IBM Proventia® Management SiteProtector® database. The maximum number of rows that the JDBC - SiteProtector protocol can poll in a single query is 30,000 rows.

The following table describes the protocol-specific parameters for the JDBC - SiteProtector protocol:

Table 3. JDBC - SiteProtector protocol parameters

Parameter	Description
Protocol Configuration	**JDBC - SiteProtector**
Database Type	From the list, select **MSDE** as the type of database to use for the event source.
Database Name	Type `RealSecureDB` the name of the database to which the protocol can connect.
IP or Hostname	The IP address or host name of the database server.
Port	The port number that is used by the database server. The JDBC SiteProtector configuration port must match the listener port of the database. The database must have incoming TCP connections enabled. If you define a **Database Instance** when with MSDE as the database type, you must leave the **Port** parameter blank in your log source configuration.
Username	If you want to track access to a database by the JDBC protocol, you can create a specific use for your QRadar system.
Authentication Domain	If you select MSDE and the database is configured for Windows, you must define a Windows domain. If your network does not use a domain, leave this field blank.

Table 3. JDBC - SiteProtector protocol parameters (continued)

Parameter	Description
Database Instance	If you select MSDE and you have multiple SQL server instances on one server, define the instance to which you want to connect. If you use a non-standard port in your database configuration, or access is blocked to port 1434 for SQL database resolution, you must leave the **Database Instance** parameter blank in your configuration.
Predefined Query	The predefined database query for your log source. Predefined database queries are only available for special log source connections.
Table Name	`SensorData1`
AVP View Name	`SensorDataAVP`
Response View Name	`SensorDataResponse`
Select List	Type `*` to include all fields from the table or view.
Compare Field	`SensorDataRowID`
Use Prepared Statements	Prepared statements allow the JDBC protocol source to set up the SQL statement, and then execute the SQL statement numerous times with different parameters. For security and performance reasons, use prepared statements. You can clear this check box to use an alternative method of querying that does not use pre-compiled statements.
Include Audit Events	Specifies to collect audit events from IBM SiteProtector®.
Start Date and Time	Optional. A start date and time for when the protocol can start to poll the database.
Polling Interval	The amount of time between queries to the event table. You can define a longer polling interval by appending H for hours or M for minutes to the numeric value. Numeric values without an H or M designator poll in seconds.
EPS Throttle	The number of Events Per Second (EPS) that you do not want this protocol to exceed.
Database Locale	For multilingual installations, use the **Database Locale** field to specify the language to use.
Database Codeset	For multilingual installations, use the **Codeset** field to specify the character set to use.
Use Named Pipe Communication	If you select MSDE as the database type, select the check box to use an alternative method to a TCP/IP port connection. When you use a Named Pipe connection, the user name and password must be the appropriate Windows authentication username and password and not the database user name and password. The log source configuration must use the default named pipe.
Database Cluster Name	The cluster name to ensure that named pipe communications function properly.
Use NTLMv2	Forces MSDE connections to use the NTLMv2 protocol with SQL servers that require NTLMv2 authentication. The **Use NTLMv2** check box does not interrupt communications for MSDE connections that do not require NTLMv2 authentication.
Use SSL	Enables SSL encryption for the JDBC protocol.

Table 3. JDBC - SiteProtector protocol parameters (continued)

Parameter	Description
Log Source Language	Select the language of the events that are generated by the log source. The log source language helps the system parse events from external appliances or operating systems that can create events in multiple languages.

Sophos Enterprise Console JDBC protocol configuration options

To receive events from Sophos Enterprise Consoles, configure a log source to use the Sophos Enterprise Console JDBC protocol.

The Sophos Enterprise Console JDBC protocol combines payload information from application control logs, device control logs, data control logs, tamper protection logs, and firewall logs in the vEventsCommonData table. If the Sophos Enterprise Console does not have the Sophos Reporting Interface, you can use the standard JDBC protocol to collect antivirus events.

The following table describes the parameters for the Sophos Enterprise Console JDBC protocol:

Table 4. Sophos Enterprise Console JDBC protocol parameters

Parameter	Description
Protocol Configuration	**Sophos Enterprise Console JDBC**
Database Type	**MSDE**
Database Name	The database name must match the database name that is specified in the **Log Source Identifier** field.
Port	The default port for MSDE in Sophos Enterprise Console is 1168. The JDBC configuration port must match the listener port of the Sophos database to communicate with QRadar. The Sophos database must have incoming TCP connections enabled. If a **Database Instance** is used with the MSDE database type, you must leave the **Port** parameter blank.
Authentication Domain	If your network does not use a domain, leave this field blank.
Database Instance	The database instance, if required. MSDE databases can include multiple SQL server instances on one server. When a non-standard port is used for the database or administrators block access to port 1434 for SQL database resolution, the **Database Instance** parameter must be blank.
Table Name	`vEventsCommonData`
Select List	`*`
Compare Field	`InsertedAt`

Table 4. Sophos Enterprise Console JDBC protocol parameters (continued)

Parameter	Description
Use Prepared Statements	Prepared statements enable the protocol source to set up the SQL statement, and then run the SQL statement numerous times with different parameters. For security and performance reasons, most configurations can use prepared statements. Clear this check box to use an alternative method of querying that do not use pre-compiled statements.
Start Date and Time	Optional. A start date and time for when the protocol can start to poll the database. If a start time is not defined, the protocol attempts to poll for events after the log source configuration is saved and deployed.
Polling Interval	The polling interval, which is the amount of time between queries to the database. You can define a longer polling interval by appending H for hours or M for minutes to the numeric value. The maximum polling interval is 1 week in any time format. Numeric values without an H or M designator poll in seconds.
EPS Throttle	The number of Events Per Second (EPS) that you do not want this protocol to exceed.
Use Named Pipe Communication	If MSDE is configured as the database type, administrators can select this check box to use an alternative method to a TCP/IP port connection. Named pipe connections for MSDE databases require the user name and password field to use a Windows authentication username and password and not the database user name and password. The log source configuration must use the default named pipe on the MSDE database.
Database Cluster Name	If you use your SQL server in a cluster environment, define the cluster name to ensure that named pipe communications function properly.
Use NTLMv2	Forces MSDE connections to use the NTLMv2 protocol with SQL servers that require NTLMv2 authentication. The default value of the check box is selected. The **Use NTLMv2** check box does not interrupt communications for MSDE connections that do not require NTLMv2 authentication.

Juniper Networks NSM protocol configuration options

To receive Juniper Networks NSM and Juniper Networks Secure Service Gateway (SSG) logs events, configure a log source to use the Juniper Networks NSM protocol.

The following table describes the protocol-specific parameters for the Juniper Networks Network and Security Manager protocol:

Table 5. Juniper Networks NSM protocol parameters

Parameter	Description
Log Source Type	**Juniper Networks Network and Security Manager**
Protocol Configuration	**Juniper NSM**

OPSEC/LEA protocol configuration options

To receive events on port 18484, configure a log source to use the OPSEC/LEA protocol is a protocol.

The following table describes the protocol-specific parameters for the OPSEC/LEA protocol:

Table 6. OPSEC/LEA protocol parameters

Parameter	Description
Protocol Configuration	**OPSEC/LEA**
Server Port	You must verify that QRadar can communicate on port 18184 by using the OPSEC/LEA protocol.
Statistics Report Interval	The interval, in seconds, during which the number of syslog events are recorded in the `qradar.log` file.
OPSEC Application Object SIC Attribute (SIC Name)	The Secure Internal Communications (SIC) name is the distinguished name (DN) of the application, for example: `CN=LEA, o=fwconsole..7psasx`.
Log Source SIC Attribute (Entity SIC Name)	The SIC name of the server, for example: `cn=cp_mgmt,o=fwconsole..7psasx`.
OPSEC Application	The name of the application that makes the certificate request.

SDEE protocol configuration options

You can configure a log source to use the Security Device Event Exchange (SDEE) protocol. QRadar uses the protocol to collect events from appliances that use SDEE servers.

The following table describes the protocol-specific parameters for the SDEE protocol:

Table 7. SDEE protocol parameters

Parameter	Description
Protocol Configuration	**SDEE**
URL	The HTTP or HTTPS URL that is required to access the log source, for example, https://www.mysdeeserver.com/cgi-bin/sdee-server. For SDEE/CIDEE (Cisco IDS v5.x and later), the URL must end with `/cgi-bin/sdee-server`. Administrators with RDEP (Cisco IDS v4.x), the URL must end with `/cgi-bin/event-server`.
Force Subscription	When the check box is selected, the protocol forces the server to drop the least active connection and accept a new SDEE subscription connection for the log source.
Maximum Wait To Block For Events	When a collection request is made and no new events are available, the protocol enables an event block. The block prevents another event request from being made to a remote device that did not have any new events. This timeout is intended to conserve system resources.

SNMPv2 protocol configuration options

You can configure a log source to use the SNMPv2 protocol to receive SNMPv2 events.

The following table describes the protocol-specific parameters for the SNMPv2 protocol:

Table 8. SNMPv2 protocol parameters

Parameter	Description
Protocol Configuration	**SNMPv3**
Community	The SNMP community name that is required to access the system that contains SNMP events.
Include OIDs in Event Payload	Specifies that the SNMP event payload is constructed by using name-value pairs instead of the event payload format. When you select specific log sources from the **Log Source Types** list, OIDs in the event payload are required for processing SNMPv2 or SNMPv3 events.

SNMPv3 protocol configuration options

You can configure a log source to use the SNMPv3 protocol to receive SNMPv3 events.

The following table describes the protocol-specific parameters for the SNMPv3 protocol:

Table 9. SNMPv3 protocol parameters

Parameter	Description
Protocol Configuration	**SNMPv3**
Authentication Protocol	The algorithms to use to authenticate SNMP traps:
Include OIDs in Event Payload	Specifies that the SNMP event payload is constructed by using name-value pairs instead of the standard event payload format. When you select specific log sources from the **Log Source Types** list, OIDs in the event payload are required for processing SNMPv2 or SNMPv3 events.

Sourcefire Defense Center Estreamer protocol configuration options

To receive events from a Sourcefire Defense Center Estreamer (Event Streamer) service, configure a log source to use the Sourcefire Defense Center Estreamer protocol.

Event files are streamed to QRadar to be processed after the Sourcefire Defense Center DSM is configured.

The following table describes the protocol-specific parameters for the Sourcefire Defense Center Estreamer protocol:

Table 10. Sourcefire Defense Center Estreamer protocol parameters

Parameter	Description
Protocol Configuration	**Sourcefire Defense Center Estreamer**
Server Port	The default port that QRadar uses for Sourcefire Defense Center Estreamer is 8302.
Keystore Filename	The directory path and file name for the keystore private key and associated certificate. By default, the import script creates the keystore file in the following directory: `/opt/qradar/conf/estreamer.keystore`.
Truststore Filename	The truststore file contains the certificates that are trusted by the client. By default, the import script creates the truststore file in the following directory: `/opt/qradar/conf/estreamer.truststore`.
Request Extra Data	Select this option to request extra data from Sourcefire Defense Center Estreamer, for example, extra data includes the original IP address of an event.
Use Extended Requests	Select this option to use an alternative method for retrieving events from an eStreamer source. Extended Requests are supported on Sourcefire DefenseCenter Estreamer version 5.0 or later.

Log file protocol configuration options

To receive events from remote hosts, configure a log source to use the log file protocol.

The log file protocol is intended for systems that write daily event logs. It is not appropriate to use the log file protocol for devices that append information to their event files.

Log files are retrieved one at a time. The log file protocol can manage plain text, compressed files, or file archives. Archives must contain plain-text files that can be processed one line at a time. When the log file protocol downloads an event file, the information that is received in the file updates the **Log Activity** tab. If more information is written to the file after the download is complete, the appended information is not processed.

The following table describes the protocol-specific parameters for the Log File protocol:

Table 11. Log file protocol parameters

Parameter	Description
Protocol Configuration	**Log File**
Remote Port	If the remote host uses a non-standard port number, you must adjust the port value to retrieve events.
SSH Key File	The path to the SSH key, if the system is configured to use key authentication. When an SSH key file is used, the **Remote Password** field is ignored.

Table 11. Log file protocol parameters (continued)

Parameter	Description
Remote Directory	For FTP, if the log files are in the remote user's home directory, you can leave the remote directory blank. A blank remote directory field supports systems where a change in the working directory (CWD) command is restricted.
Recursive	This option is ignored for SCP file transfers.
FTP File Pattern	The regular expression (regex) required to identify the files to download from the remote host.
FTP Transfer Mode	For ASCII transfers over FTP, you must select `NONE` in the **Processor** field and `LINEBYLINE` in the **Event Generator** field.
Recurrence	The time interval to determine how frequently the remote directory is scanned for new event log files. The time interval can include values in hours (H), minutes (M), or days (D). For example, a recurrence of 2H scans the remote directory every 2 hours.
Run On Save	Starts the log file import immediately after you save the log source configuration. When selected, this check box clears the list of previously downloaded and processed files. After the first file import, the log file protocol follows the start time and recurrence schedule that is defined by the administrator.
EPS Throttle	The number of Events Per Second (EPS) that the protocol cannot exceed.
Change Local Directory?	Changes the local directory on the **Target Event Collector** to store event logs before they are processed.
Local Directory	The local directory on the **Target Event Collector**. The directory must exist before the log file protocol attempts to retrieve events.
File Encoding	The character encoding that is used by the events in your log file.
Folder Separator	The character that is used to separate folders for your operating system. Most configurations can use the default value in **Folder Separator** field. This field is intended for operating systems that use a different character to define separate folders. For example, periods that separate folders on mainframe systems.

Microsoft Security Event Log protocol configuration options

You can configure a log source to use the Microsoft Security Event Log protocol. You can use Microsoft Windows Management Instrumentation (WMI) to collect customized event logs or agent less Windows Event Logs.

The WMI API requires that firewall configurations accept incoming external communications on port 135 and on any dynamic ports that are required for DCOM. The following list describes the log source limitations that you use the Microsoft Security Event Log Protocol:

- Systems that exceed 50 events per second (eps) might exceed the capabilities of this protocol. Use WinCollect for systems that exceed 50 eps.

- A QRadar all-in-one installation can support up to 250 log sources with the Microsoft Security Event Log protocol.
- Dedicated Event Collectors can support up to 500 log sources by using the Microsoft Security Event Log protocol.

The Microsoft Security Event Log protocol is not suggested for remote servers that are accessed over network links, for example, systems that have high round-trip delay times, such as satellite or slow WAN networks. You can confirm round-trip delays by examining requests and response time that is between a server ping. Network delays that are created by slow connections decrease the EPS throughput available to those remote servers. Also, event collection from busy servers or domain controllers rely on low round-trip delay times to keep up with incoming events. If you cannot decrease your network round-trip delay time, you can use WinCollect to process Windows events.

The Microsoft Security Event Log supports the following software versions with the Microsoft Windows Management Instrumentation (WMI) API:

- Microsoft Windows 2000
- Microsoft Windows Server 2003
- Microsoft Windows Server 2008
- Microsoft Windows Server 2008R3
- Microsoft Windows XP
- Microsoft Windows Vista
- Microsoft Windows 7

The following table describes the protocol-specific parameters for the Microsoft Security Event Log protocol:

Table 12. Microsoft Security Event Log protocol parameters

Parameter	Description
Protocol Configuration	**Windows Security Event Log**

Microsoft DHCP protocol configuration options

To receive events from Microsoft DHCP servers, configure a log source to use the Microsoft DHCP protocol.

To read the log files, folder paths that contain an administrative share (C$), require NetBIOS privileges on the administrative share (C$). Local or domain administrators have sufficient privileges to access log files on administrative shares.

Fields for the Microsoft DHCP protocol that support file paths allow administrators to define a drive letter with the path information. For example, the field can contain the `c$/LogFiles/` directory for an administrative share, or the `LogFiles/`directory for a public share folder path, but cannot contain the `c:/LogFiles` directory.

Restriction: The Microsoft authentication protocol NTLMv2 is not supported by the Microsoft DHCP protocol.

The following table describes the protocol-specific parameters for the Microsoft DHCP protocol:

Table 13. Microsoft DHCP protocol parameters

Parameter	Description
Protocol Configuration	**Microsoft DHCP**
Domain	Optional.
Folder Path	The directory path to the DHCP log files.
File Pattern	The regular expression (regex) that identifies event logs. The log files must contain a three-character abbreviation for a day of the week. Use one of the following file patterns: • IPv4 file pattern: `DhcpSrvLog-(?:Sun|Mon|Tue|Wed|Thu|Fri| Sat)\.log`. • IPv6 file pattern: `DhcpV6SrvLog-(?:Sun|Mon|Tue|Wed|Thu|Fri|Sat) \.log`. • Mixed IPv4 and IPv6 file pattern: `Dhcp.*SrvLog-(?:Sun|Mon|Tue|Wed|Thu|Fri|Sat) \.log`.

Microsoft Exchange protocol configuration options

To receive events from SMTP, OWA, and Microsoft Exchange 2007 and 2010 servers, configure a log source to use the Microsoft Windows Exchange protocol to support.

To read the log files, folder paths that contain an administrative share (C$), require NetBIOS privileges on the administrative share (C$). Local or domain administrators have sufficient privileges to access log files on administrative shares.

Fields for the Microsoft Exchange protocol that support file paths allow administrators to define a drive letter with the path information. For example, the field can contain the `c$/LogFiles/` directory for an administrative share, or the `LogFiles/`directory for a public share folder path, but cannot contain the `c:/LogFiles` directory.

Important: The Microsoft Exchange protocol does not support Microsoft Exchange 2003 or Microsoft authentication protocol NTLMv2 Session.

The following table describes the protocol-specific parameters for the Microsoft Exchange protocol:

Table 14. Microsoft Exchange protocol parameters

Parameter	Description
Protocol Configuration	**Microsoft Exchange**
Domain	Optional.
SMTP Log Folder Path	When the folder path is clear, SMTP event collection is disabled.
OWA Log Folder Path	When the folder path is clear, OWA event collection is disabled.
MSGTRK Log Folder Path	Message tracking is available on Microsoft Exchange 2007 or 2010 servers assigned the Hub Transport, Mailbox, or Edge Transport server role.
File Pattern	The regular expression (regex) that identifies the event logs. The default is `.*\.(?:log|LOG)`.

Table 14. Microsoft Exchange protocol parameters (continued)

Parameter	Description
Force File Read	If the check box is cleared, the log file is read only when QRadar detects a change in the modified time or file size.
Throttle Events/Second	The maximum number of events the Exchange protocol can forward per second.

Microsoft IIS protocol configuration options

You can configure a log source to use the Microsoft IIS protocol. This protocol supports a single point of collection for W3C format log files that are located on a Microsoft IIS web server.

To read the log files, folder paths that contain an administrative share (C$), require NetBIOS privileges on the administrative share (C$). Local or domain administrators have sufficient privileges to access log files on administrative shares.

Fields for the Microsoft IIS protocol that support file paths allow administrators to define a drive letter with the path information. For example, the field can contain the `c$/LogFiles/` directory for an administrative share, or the `LogFiles/`directory for a public share folder path, but cannot contain the `c:/LogFiles` directory.

Restriction: The Microsoft authentication protocol NTLMv2 is not supported by the Microsoft IIS protocol.

The following table describes the protocol-specific parameters for the Microsoft IIS protocol:

Table 15. Microsoft IIS protocol parameters

Parameter	Description
Protocol Configuration	**Microsoft IIS**
File Pattern	The regular expression (regex) that identifies the event logs.
Throttle Events/Second	The maximum number of events the IIS protocol can forward per second.

SMB Tail protocol configuration options

You can configure a log source to use the SMB Tail protocol. Use this protocol to watch events on a remote Samba share and receive events from the Samba share when new lines are added to the event log.

The following table describes the protocol-specific parameters for the SMB Tail protocol:

Table 16. SMB Tail protocol parameters

Parameter	Description
Protocol Configuration	**SMB Tail**

Table 16. SMB Tail protocol parameters (continued)

Parameter	Description
Log Folder Path	The directory path to access the log files. For example, administrators can use the `c$/LogFiles/` directory for an administrative share, or the `LogFiles/` directory for a public share folder path. However, the `c:/LogFiles` directory is not a supported log folder path. If a log folder path contains an administrative share (C$), users with NetBIOS access on the administrative share (C$) have the privileges that are required to read the log files. Local system or domain administrator privileges are also sufficient to access a log files that are on an administrative share.
File Pattern	The regular expression (regex) that identifies the event logs.
Force File Read	If the check box is cleared, the log file is read only when QRadar detects a change in the modified time or file size.
Throttle Events/Second	The maximum number of events the SMB Tail protocol forwards per second.

EMC VMware protocol configuration options

To receive event data from the VMWare web service for virtual environments, configure a log source to use the EMC VMWare protocol.

The following table describes the protocol-specific parameters for the EMC VMware protocol:

Table 17. EMC VMware protocol parameters

Parameter	Description
Protocol Configuration	**EMC VMware**
Log Source Identifier	The value for this parameter must match the **VMware IP** parameter.
VMware IP	The IP address of the VMWare ESXi server, for example, `1.1.1.1`. The VMware protocol appends the IP address of your VMware ESXi server with HTTPS before the protocol requests event data.

Oracle Database Listener protocol configuration options

To remotely collect log files that are generated from an Oracle database server, configure a log source to use the Oracle Database Listener protocol source.

Before you configure the Oracle Database Listener protocol to monitor log files for processing, you must obtain the directory path to the Oracle database log files.

The following table describes the protocol-specific parameters for the Oracle Database Listener protocol:

Table 18. Oracle Database Listener protocol parameters

Parameter	Description
Protocol Configuration	**Oracle Database Listener**

Table 18. Oracle Database Listener protocol parameters (continued)

Parameter	Description
File Pattern	The regular expression (regex) that identifies the event logs.

Cisco NSEL protocol configuration options

To monitor NetFlow packet flows from a Cisco Adaptive Security Appliance (ASA), configure the Cisco Network Security Event Logging (NSEL) protocol source.

To integrate Cisco NSEL with QRadar, you must manually create a log source to receive NetFlow events. QRadar does not automatically discover or create log sources for syslog events from Cisco NSEL. For more information, see the *DSM Configuration Guide*.

The following table describes the protocol-specific parameters for the Cisco NSEL protocol:

Table 19. Cisco NSEL protocol parameters

Parameter	Description
Protocol Configuration	**Cisco NSEL**
Log Source Identifier	If the network contains devices that are attached to a management console, you can specify the IP address of the individual device that created the event. A unique identifier for each, such as an IP address, prevents event searches from identifying the management console as the source for all of the events.
Collector Port	The UDP port number that Cisco ASA uses to forward NSEL events. QRadar uses port 2055 for flow data on QRadar QFlow Collectors. You must assign a different UDP port on the Cisco Adaptive Security Appliance for NetFlow.

PCAP Syslog Combination protocol configuration options

To collect events from Juniper Networks SRX Series appliances that forward packet capture (PCAP) data, configure a log source to use the PCAP Syslog Combination protocol .

Before you configure a log source that uses the PCAP Syslog Combination protocol, determine the outgoing PCAP port that is configured on the Juniper Networks SRX appliance. PCAP data cannot be forwarded to port 514.

The following table describes the protocol-specific parameters for the PCAP Syslog Combination protocol:

Table 20. PCAP Syslog Combination protocol parameters

Parameter	Description
Protocol Configuration	**PCAP Syslog Combination**
Incoming PCAP Port	If the outgoing PCAP port is edited on the Juniper Networks SRX Series appliance, you must edit the log source to update the incoming PCAP Port. After you edit the **Incoming PCAP Port** field, you must deploy your changes.

Forwarded protocol configuration options

To receive events from another Console in your deployment, configure a log source to use the Forwarded protocol.

The Forwarded protocol is typically used to forward events to another QRadar Console. For example, Console A has Console B configured as an off-site target. Data from automatically discovered log sources is forwarded to Console B. Manually created log sources on Console A must also be added as a log source to Console B with the forwarded protocol.

TLS syslog protocol configuration options

To receive encrypted syslog events from up to 50 network devices that support TLS Syslog event forwarding, configure a log source to use the TLS Syslog protocol.

The log source creates a listen port for incoming TLS Syslog events and generates a certificate file for the network devices. Up to 50 network appliances can forward events to the listen port that is created for the log source. If you require more than 50 network appliances, create additional listen ports.

The following table describes the protocol-specific parameters for the TLS Syslog protocol:

Table 21. TLS syslog protocol parameters

Parameter	Description
Protocol Configuration	**TLS Syslog**
TLS Listen Port	The default TLS listen port is 6514.
Authentication Mode	The mode by which your TLS connection is authenticated. If you select the **TLS and Client Authentication** option, you must configure the certificate parameters.
Client Certificate Path	The absolute path to the client-certificate on disk. The certificate must be stored on the Console or Event Collector for this log source.
Certificate Type	The type of certificate to use for authentication. If you select the **Provide Certificate** option, you must configure the file paths for the server certificate and the private key.
Provided Server Certificate Path	The absolute path to the server certificate.
Provided Private Key Path	The absolute path to the private key. **Note:** The corresponding private key must be a DER-encoded PKCS8 key. The configuration fails with any other key format.

TLS syslog use cases

The following use cases represent possible configurations that you can create:

Client Authentication

You can supply a client-certificate that enables the protocol to engage in client-authentication. If you select this option and provide the certificate, incoming connections are validated against the client-certificate.

User-provided Server Certificates
: You can configure your own server certificate and corresponding private key. The configured TLS Syslog provider uses the certificate and key. Incoming connections are presented with the user-supplied certificate, rather than the automatically generated TLS Syslog certificate.

Default authentication
: To use the default authentication method, use the default values for the **Authentication Mode** and **Certificate Type** parameters. After the log source is saved, a `syslog-tls` certificate is created for log source device. The certificate must be copied to any device on your network that forwards encrypted syslog data.

Juniper Security Binary Log Collector protocol configuration options

You can configure a log source to use the Security Binary Log Collector protocol. With this protocol, Juniper appliances can send audit, system, firewall, and intrusion prevention system (IPS) events in binary format to QRadar.

The binary log format from Juniper SRX or J Series appliances are streamed by using the UDP protocol. You must specify a unique port for streaming binary formatted events. The standard syslog port 514 cannot be used for binary formatted events. The default port that is assigned to receive streaming binary events from Juniper appliances is port 40798.

The following table describes the protocol-specific parameters for the Juniper Security Binary Log Collector protocol:

Table 22. Juniper Security Binary Log Collector protocol parameters

Parameter	Description
Protocol Configuration	**Security Binary Log Collector**
XML Template File Location	The path to the XML file used to decode the binary stream from your Juniper SRX or Juniper J Series appliance. By default, the device support module (DSM) includes an XML file for decoding the binary stream. The XML file is in the following directory: `/opt/qradar/conf/security_log.xml`.

UDP multiline syslog protocol configuration options

To create a single-line syslog event from a multiline event, configure a log source to use the UDP multiline protocol. The UDP multiline syslog protocol uses a regular expression to identify and reassemble the multiline syslog messages into single event payload.

The original event must contain a value that repeats a regular expression that can identify and reassemble the multiline event. For example, this event contains a repeated value:

```
15:08:56 1.1.1.1 slapd[517]: conn=2467222 op=2 SEARCH RESULT tag=101
15:08:56 1.1.1.1 slapd[517]: conn=2467222 op=2 SRCH base="dc=iso-n,dc=com"
15:08:56 1.1.1.1 slapd[517]: conn=2467222 op=2 SRCH attr=gidNumber
15:08:56 1.1.1.1 slapd[517]: conn=2467222 op=1 SRCH base="dc=iso-n,dc=com"
```

The following table describes the protocol-specific parameters for the UDP multiline syslog protocol:

Table 23. UDP multiline syslog protocol parameters

Parameter	Description
Protocol Configuration	**UDP Multiline Syslog**
Message ID Pattern	The regular expression (regex) required to filter the event payload messages. The UDP multiline event messages must contain a common identifying value that repeats on each line of the event message.

After the log source is saved, a syslog-tls certificate is created for the log source. The certificate must be copied to any device on your network that is configured to forward encrypted syslog. Other network devices that have a syslog-tls certificate file and the TLS listen port number can be automatically discovered as a TLS syslog log source.

TCP multiline syslog protocol configuration options

You can configure a log source that uses the TCP multiline syslog protocol. To create a single-line event, this protocol uses regular expressions to identify the start and end pattern of multiline events.

The following example is a multiline event:

```
06/13/2012 08:15:15 PM
LogName=Security
SourceName=Microsoft Windows security auditing.
EventCode=5156
EventType=0
TaskCategory=Filtering Platform Connection
Keywords=Audit Success
Message=The Windows Filtering Platform permitted a connection.
Process ID: 4
Application Name: System
Direction: Inbound
Source Address: 1.1.1.1
Source Port: 80
Destination Address: 1.1.1.12
Destination Port:444
```

The following table describes the protocol-specific parameters for the TCP multiline syslog protocol:

Table 24. TCP multiline syslog protocol parameters

Parameter	Description
Protocol Configuration	**TCP Multiline Syslog**
Listen Port	The default listen port is 12468.
Event Formatter	Use the **Windows Multiline** option for multiline events that are formatted specifically for Windows.
Event Start Pattern	The regular expression (regex) that is required to identify the start of a TCP multiline event payload. Syslog headers typically begin with a date or time stamp. The protocol can create a single-line event that is based on solely an event start pattern, such as a time stamp. When only a start pattern is available, the protocol captures all the information between each start value to create a valid event.

Table 24. TCP multiline syslog protocol parameters (continued)

Parameter	Description
Event End Pattern	The regular expression (regex) that is required to identify the last field of a TCP multiline event payload. If the syslog event ends with the same value, you can use a regular expression to determine the end of an event. The protocol can capture events that are based on solely on an event end pattern. When only an end pattern is available, the protocol captures all the information between end start value to create a valid event.

VMware vCloud Director protocol configuration options

To collect events from the VMware vCloud Director virtual environments, you can create a log source that uses the VMware vCloud Director protocol.

The following table describes the protocol-specific parameters for the VMware vCloud Director protocol:

Table 25. VMware vCloud Director protocol parameters

Parameter	Description
Protocol Configuration	**VMware vCloud Director**
vCloud URL	The URL that is configured on the VMware vCloud appliance to access the REST API. The URL must match the address that is configured as the VCD public REST API base URL on the vCloud Server, for example, `https://1.1.1.1.`.
User Name	The user name that is required to remotely access the vCloud Server, for example, `console/user@organization`. To configure a read-only account to use with the vCloud Director protocol, a user must have Console Access Only permission.

IBM Tivoli Endpoint Manager SOAP protocol configuration options

To receive Log Extended Event Format (LEEF) formatted events from IBM Tivoli® Endpoint Manager appliances, configure a log source that uses the IBM Tivoli Endpoint Manager SOAP protocol.

This protocol requires IBM Tivoli Endpoint Manager versions V8.2.x or later and the Web Reports application for Tivoli Endpoint Manager.

The Tivoli Endpoint Manager SOAP protocol retrieves events in 30-second intervals over HTTP or HTTPS. As events are retrieved, the IBM Tivoli Endpoint Manager DSM parses and categorizes the events.

The following table describes the protocol-specific parameters for the IBM Tivoli Endpoint Manager SOAP protocol:

Table 26. IBM Tivoli Endpoint Manager SOAP protocol parameters

Parameter	Description
Protocol Configuration	**IBM Tivoli Endpoint Manager SOAP**

Table 26. IBM Tivoli Endpoint Manager SOAP protocol parameters (continued)

Parameter	Description
Use HTTPS	If a certificate is required to connect with HTTPS, copy the required certificates to the following directory: `/opt/qradar/conf/trusted_certificates`. Certificates that have following file extensions: `.crt`, `.cert`, or `.der` are supported. Copy the certificates to the trusted certificates directory before the log source is saved and deployed.
SOAP Port	By default, port 80 is the port number for communicating with IBM Tivoli Endpoint Manager. Most configurations use port 443 for HTTPS communications.

Syslog Redirect protocol overview

The Syslog Redirect protocol is used as an alternative to the Syslog protocol. Use this protocol when you want to QRadar identify the specific device name that sent the events. QRadar can passively listen for Syslog events on UDP port 517.

The following table describes the protocol-specific parameters for the Syslog Redirect protocol:

Table 27. Syslog Redirect protocol parameters

Parameter	Description
Protocol Configuration	**Syslog Redirect**
Log Source Identifier RegEx	`devname=([\w-]+)`
Listen Port	`517`
Protocol	**UDP**

Adding bulk log sources

You can add up to 500 Microsoft Windows or Universal DSM log sources at one time. When you add multiple log sources at one time, you add a bulk log source in QRadar. Bulk log sources must share a common configuration.

Procedure

1. Click the **Admin** tab.
2. Click the **Log Sources** icon.
3. From the **Bulk Actions** list, select **Bulk Add**.
4. Configure the parameters for the bulk log source.
   - File Upload - Upload a text file that has one host name or IP per line
   - Manual - Enter the host name or IP of the host that you wish to add
5. Click **Save**.
6. Click **Continue** to add the log sources.
7. On the **Admin** tab, click **Deploy Changes**.

Adding a log source parsing order

You can assign a priority order for when the events are parsed by the target event collector.

About this task

You can order the importance of the log sources by defining the parsing order for log sources that share a common IP address or host name. Defining the parsing order for log sources ensures that certain log sources are parsed in a specific order, regardless of changes to the log source configuration. The parsing order ensures that system performance is not affected by changes to log source configuration by preventing unnecessary parsing. The parsing order ensures that low-level event sources are not parsed for events before more important log source.

Procedure

1. Click the **Admin** tab.
2. Click the **Log Source Parsing Ordering** icon.
3. Select a log source.
4. Optional: From the **Selected Event Collector** list, select the Event Collector to define the log source parsing order.
5. Optional: From the **Log Source Host** list, select a log source.
6. Prioritize the log source parsing order.
7. Click **Save**.

Chapter 2. Log source extensions

An extension document can extend or modify how the elements of a particular log source are parsed. You can use the extension document correct a parsing issue or override the default parsing for an event from an existing DSM.

An extension document can also provide event support when a DSM does not exist to parse events for an appliance or security device in your network.

An extension document is an Extensible Markup Language (XML) formatted document that you can create or edit one by using any common text, code or markup editor. You can create multiple extension documents but a log source can have only one applied to it.

The XML format requires that all regular expression (regex) patterns be contained in character data (CDATA) sections to prevent the special characters that are required by regular expressions from interfering with the markup format. For example, the following code shows the regex for finding protocols:

```
<pattern id="ProtocolPattern" case-insensitive="true" xmlns="">
<![CDATA[(TCP|UDP|ICMP|GRE)]]></pattern>
```

`(TCP|UDP|ICMP|GRE)` is the regular expression pattern.

The log sources extension configuration consists of the following sections:

Pattern
: Regular expressions patterns that you associate with a particular field name. Patterns are referenced multiple times within the log source extension file.

Match groups
: An entity within a match group that is parsed, for example, EventName, and is paired with the appropriate pattern and group for parsing. Any number of match groups can appear in the extension document.

Examples of log source extensions on QRadar forum

You can create log source extensions (LSX) for log sources that don't have a supported DSM. To help you create your own log source extensions (also known as DSM extensions), you modify existing ones that were created.

You can access log source extension examples (https://www.ibm.com/developerworks/community/forums/html/topic?id=d15cac8d-b0fa-4461-bb1e-dc1b291de440&ps=25) on the Discussion about DSM Extensions, Custom Properties and other REGEX related topics forum (https://www.ibm.com/developerworks/community/forums/html/forum?id=11111111-0000-0000-0000-000000003046&ps=25).

The IBM Security QRadar forums is an online discussion site where users and subject matter experts collaborate and share information.

Related concepts:

"Creating a log source extensions document" on page 33
Create log source extensions (LSX) for log sources that don't have a supported DSM, or to repair an event that has missing or incorrect information, or to parse an event when the associated DSM fails to produce a result.

Patterns in log source extension documents

Rather than associating a regular expression directly with a particular field name, patterns (patterns) are declared separately at the top of the extension document. These regex patterns can be then referenced multiple times within the log source extension file.

All characters between the start tag <pattern> and end tag </pattern> are considered part of the pattern. Do not use extra spaces or hard returns inside or around your pattern or <CDATA> expression. Extra characters or spaces can prevent the DSM extension from matching your intended pattern.

Table 28. Description of pattern parameters

Pattern	Type	Description
id (Required)	String	A regular string that is unique within the extension document.
case-insensitive (Optional)	Boolean	If true, the character case is ignored. For example, abc is the same as ABC. If not specified, this parameter defaults to false.
trim-whitespace (Optional)	Boolean	If true, whitespace and carriage returns are ignored. If the CDATA sections are split onto different lines, any extra spaces and carriage returns are not interpreted as part of the pattern. If not specified, this parameter defaults to false.

Match groups

A *match group* (match-group) is a set of patterns that are used for parsing or modifying one or more types of events.

A *matcher* is an entity within a match group that is parsed, for example, EventName, and is paired with the appropriate pattern and group for parsing. Any number of match groups can appear in the extension document.

Table 29. Description of match group parameters

Parameter	Description
order (Required)	An integer greater than zero that defines the order in which the match groups are executed. It must be unique within the extension document.

Table 29. Description of match group parameters (continued)

Parameter	Description
`description` (Optional)	A description for the match group, which can be any string. This information can appear in the logs. If not specified, this parameter defaults to empty.
`device-type-id-override` (Optional)	Define a different device ID to override the QID. Allows the particular match group to search in the specified device for the event type. It must be a valid log source type ID, represented as an integer. A list of log source type IDs is presented in Table 36 on page 44. If not specified, this parameter defaults to the log source type of the log source to which the extension is attached.

Match groups can have these entities:

- "Matcher (matcher)"
- "Single-event modifier (`event-match-single`)" on page 29
- "Multi-event modifier (`event-match-multiple`)" on page 29

Matcher (matcher)

A matcher entity is a field that is parsed, for example, EventName, and is paired with the appropriate pattern and group for parsing.

Matchers have an associated order. If multiple matchers are specified for the same field name, the matchers are run in the order that is presented until a successful parse is found or a failure occurs.

Table 30. Description of matcher parameters

Parameter	Description
`field` (Required)	The field to which you want the pattern to apply, for example, EventName, or SourceIp. You can use any of the field names that are listed in the List of valid matcher field names table.
`pattern-id` (Required)	The pattern that you want to use when the field is parsed from the payload. This value must match (including case) the ID parameter of the pattern that is previously defined in a pattern ID parameter (Table 28 on page 24).
`order` (Required)	The order that you want this pattern to attempt among matchers that are assigned to the same field. If two matchers are assigned to the EventName field, the one with the lowest order is attempted first.

Table 30. Description of matcher parameters (continued)

Parameter	Description
`capture-group` (Optional)	Referenced in the regular expression inside parenthesis ( ). These captures are indexed starting at one and processed from left to right in the pattern. The `capture-group` field must be a positive integer less than or equal to the number of capture groups that are contained in the pattern. The default value is zero, which is the entire match. For example, you can define a single pattern for a source IP address and port; where the SourceIp matcher can use a capture group of 1, and the SourcePort matcher can use a capture group of 2, but only one pattern needs to be defined. This field has a dual purpose when combined with the `enable-substitutions` parameter. To see an example, review the extension document example.

`enable-substitutions` (Optional)	Boolean
	When you set to `true`, a field cannot be adequately represented with a straight group capture. You can combine multiple groups with extra text to form a value.
	This parameter changes the meaning of the `capture-group` parameter. The `capture-group` parameter creates the new value, and group substitutions are specified by using `\x` where `x` is a group number, 1 - 9. You can use groups multiple times, and any free-form text can also be inserted into the value. For example, to form a value out of group 1, followed by an underscore, followed by group 2, an @, and then group 1 again, the appropriate capture-group syntax is shown in the following code:
	`capture-group="\1_\2@\1"`
	In another example, a MAC address is separated by colons, but in QRadar, MAC addresses are usually hyphen-separated. The syntax to parse and capture the individual portions is shown in the following example:
	`capture-group="\1:\2:\3:\4:\5:\6"`
	If no groups are specified in the capture-group when substitutions are enabled, a direct text replacement occurs.
	Default is false.

Table 30. Description of matcher parameters (continued)

Parameter	Description
`ext-data` (Optional)	An extra-data parameter that defines any extra field information or formatting that a matcher field can provide in the extension. The only field that uses this parameter is DeviceTime. For example, you might have a device that sends events by using a unique time stamp, but you want the event to be reformatted to a standard device time. Use the ext-data parameter included with the DeviceTime field to reformat the date and time stamp of the event. For more information, see the List of valid matcher field names.

The following table lists valid matcher field names.

Table 31. List of valid matcher field names

Field name	Description
EventName (Required)	The event name to be retrieved from the QID to identify the event. **Note:** This parameter doesn't appear as a field in the **Log Activity** tab.
EventCategory	An event category for any event with a category not handled by an event-match-single entity or an event-match-multiple entity. Combined with EventName, EventCategory is used to search for the event in the QID. The fields that are used for QIDmap lookups require an override flag to be set when the devices are already known to QRadar, for example, `<event-match-single event-name= "Successfully logged in" force-qidmap-lookup-on-fixup="true" device-event-category="CiscoNAC" severity="4" send-identity= "OverrideAndNeverSend" />` The `force-qidmap-lookup-on-fixup="true"` is the flag override. **Note:** This parameter doesn't appear as a field in the **Log Activity** tab.
SourceIp	The source IP address for the message.
SourcePort	The source port for the message.
SourceIpPreNAT	The source IP address for the message before Network Address Translation (NAT) occurs.
SourceIpPostNAT	The source IP address for the message after NAT occurs.

Table 31. List of valid matcher field names (continued)

Field name	Description
SourceMAC	The source MAC address for the message.
SourcePortPreNAT	The source port for the message before NAT occurs.
SourcePortPostNAT	The source port for the message after NAT occurs.
DestinationIp	The destination IP address for the message.
DestinationPort	The destination port for the message.
DestinationIpPreNAT	The destination IP address for the message before NAT occurs.
DestinationIpPostNAT	The destination IP address for the message after NAT occurs.
DestinationPortPreNAT	The destination port for the message before NAT occurs.
DestinationPortPostNAT	The destination port for the message after NAT occurs.
DestinationMAC	The destination MAC address for the message.
DeviceTime	The time and format that is used by the device. This date and time stamp represent the time that the event was sent, according to the device. This parameter doesn't represent the time that the event arrived. The `DeviceTime` field supports the ability to use a custom date and time stamp for the event by using the ext-data Matcher attribute. The following list contains examples of date and time stamp formats that you can use in the `DeviceTime` field: • ext-data="dd/MMM/YYYY:hh:mm:ss" 11/Mar/2015:05:26:00 • ext-data="MMM dd YYYY / hh:mm:ss" Mar 11 2015 / 05:26:00 • ext-data="hh:mm:ss:dd/MMM/YYYY" 05:26:00:11/Mar/2015 For more information about the possible values for the data and time stamp format, see the Joda-Time web page (http://www.joda.org/joda-time/key_format.html). DeviceTime is the only event field that uses the ext-data optional parameter.
Protocol	The protocol that is associated with the event; for example, TCP, UDP, or ICMP.
UserName	The user name that is associated with the event.

Table 31. List of valid matcher field names (continued)

Field name	Description
HostName	The host name that is associated with the event. Typically, this field is associated with identity events.
GroupName	The group name that is associated with the event. Typically, this field is associated with identity events.
NetBIOSName	The NetBIOS name that is associated with the event. Typically, this field is associated with identity events.
ExtraIdentityData	Any user-specific data that is associated with the event. Typically, this field is associated with identity events.
SourceIpv6	The IPv6 source IP address for the message.
DestinationIpv6	The IPv6 destination IP address for the message.

Multi-event modifier (`event-match-multiple`)

The multi-event modifier (`event-match-multiple`) matches a range of event types and then modifies them as specified by the `pattern-id` parameter and the `capture-group-index` parameter.

This match is not done against the payload, but is done against the results of the EventName matcher previously parsed out of the payload.

This entity allows mutation of successful events by changing the device event category, severity, or the method the event uses to send identity events. The `capture-group-index` must be an integer value (substitutions are not supported) and pattern-ID must reference an existing pattern entity. All other properties are identical to their counterparts in the single-event modifier.

Single-event modifier (`event-match-single`)

Single-event modifier (`event-match-single`) matches and then modifies exactly one type of event, as specified by the required, case-sensitive EventName parameter.

This entity allows mutation of successful events by changing the device event category, severity, or the method for sending identity events.

When events that match this event name are parsed, the device category, severity, and identity properties are imposed upon the resulting event.

You must set an `event-name` attribute and this attribute value matches the value of the **EventName** field. In addition, an event-match-single entity consists of these optional properties:

Table 32. Description of single-event parameters

Parameter	Description
device-event-category	A new category for searching for a QID for the event. This parameter is an optimizing parameter because some devices have the same category for all events.
severity	The severity of the event. This parameter must be an integer value 1 - 10. If a severity of less than 1 or greater than 10 is specified, the system defaults to 5. If not specified, the default is whatever is found in the QID.
send-identity	Specifies the sending of identity change information from the event. Choose one of the following options: • UseDSMResults If the DSM returns an identity event, the event is passed on. If the DSM does not return an identity event, the extension does not create or modify the identity information. This option is the default value if no value is specified. • SendIfAbsent If the DSM creates identity information, the identity event is passed through unaffected. If no identity event is produced by the DSM, but there is enough information in the event to create an identity event, an event is generated with all the relevant fields set. • OverrideAndAlwaysSend Ignores any identity event that is returned by the DSM and creates a new identity event, if there is enough information. • OverrideAndNeverSend Suppress any identity information that is returned by the DSM. Suggested option unless you are processing events that you want to go into asset updates.

Extension document template

The example of an extension document provides information about how to parse one particular type of Cisco FWSM so that events are not sent with an incorrect event name.

For example, if you want to resolve the word session, which is embedded in the middle of the event name:

```
Nov 17 09:28:26 129.15.126.6 %FWSM-session-0-302015:
Built UDP connection for faddr 38.116.157.195/80
gaddr 129.15.127.254/31696 laddr 10.194.2.196/2157
duration 0:00:00 bytes 57498 (TCP FINs)
```

This condition causes the DSM to not recognize any events and all the events are unparsed and associated with the generic logger.

Although only a portion of the text string (`302015`) is used for the QID search, the entire text string (`%FWSM-session-0-302015`) identifies the event as coming from a Cisco FWSM. Since the entire text string is not valid, the DSM assumes that the event is not valid.

Extension document example for parsing one event type

An FWSM device has many event types and many with unique formats. The following extension document example indicates how to parse one event type.

Note: The pattern IDs do not have to match the field names that they are parsing. Although the following example duplicates the pattern, the `SourceIp` field and the `SourceIpPreNAT` field cab use the exact same pattern in this case. This situation might not be true in all FWSM events.

```
<?xml version="1.0" encoding="UTF-8"?>
<device-extension xmlns="event_parsing/device_extension">
<pattern id="EventNameFWSM_Pattern" xmlns=""><![CDATA[%FWSM[a-zA-Z\-]*\d-(\d{1,6})]]></pattern>
<pattern id="SourceIp_Pattern" xmlns=""><![CDATA[gaddr (\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})/([\d]{1,5})]]></pattern>
<pattern id="SourceIpPreNAT_Pattern" xmlns=""><![CDATA[gaddr (\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})/([\d]{1,5})]]></pattern>
<pattern id="SourceIpPostNAT_Pattern" xmlns=""><![CDATA[laddr (\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})/([\d]{1,5})]]></pattern>
<pattern id="DestinationIp_Pattern" xmlns=""><![CDATA[faddr (\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})/([\d]{1,5})]]></pattern>
<pattern id="Protocol_Pattern" case-insensitive="true" xmlns=""><![CDATA[(tcp|udp|icmp|gre)]]></pattern>
<pattern id="Protocol_6_Pattern" case-insensitive="true" xmlns=""><![CDATA[protocol=6]]></pattern>
<pattern id="EventNameId_Pattern" xmlns=""><![CDATA[(\d{1,6})]]></pattern>
<match-group order="1" description="FWSM Test" device-type-id-override="6" xmlns="">
      <matcher field="EventName" order="1" pattern-id="EventNameFWSM_Pattern" capture-group="1"/>
      <matcher field="SourceIp" order="1" pattern-id="SourceIp_Pattern" capture-group="1" />
      <matcher field="SourcePort" order="1" pattern-id="SourcePort_Pattern" capture-group="2"/>
      <matcher field="SourceIpPreNAT" order="1" pattern-id="SourceIpPreNAT_Pattern" capture-group="1" />
      <matcher field="SourceIpPostNAT" order="1" pattern-id="SourceIpPostNAT_Pattern" capture-group="1" />
      <matcher field="SourcePortPreNAT" order="1" pattern-id="SourcePortPreNAT_Pattern" capture-group="2" />
      <matcher field="SourcePortPostNAT" order="1" pattern-id="SourcePortPostNAT_Pattern" capture-group="2" />
      <matcher field="DestinationIp" order="1" pattern-id="DestinationIp_Pattern" capture-group="1" />
      <matcher field="DestinationPort" order="1" pattern-id="DestinationIp_Pattern" capture-group="2" />
      <matcher field="Protocol" order="1" pattern-id="Protocol_Pattern" capture-group="1" />
      <matcher field="Protocol" order="2" pattern-id="Protocol_6_Pattern" capture-group="TCP" enable-substitutions=true/>
      <event-match-multiple pattern-id="EventNameId" capture-group-index="1" device-event-category="Cisco Firewall"/>
   </match-group>
</device-extension>
```

```
<?xml version="1.0" encoding="UTF-8"?>
<device-extension xmlns="event_parsing/device_extension">
<!-- Do not remove the "allEventNames" value -->
<pattern id="EventName-Fakeware_Pattern" xmlns=""><![CDATA[]]></pattern>
<pattern id="SourceIp-Fakeware_Pattern" xmlns=""><![CDATA[]]</pattern>
<pattern id="SourcePort-Fakeware_Pattern" xmlns=""><![CDATA[]]></pattern>
<pattern id="SourceMAC-Fakeware_Pattern" xmlns=""><![CDATA[]]></pattern>
<pattern id="DestinationIp-Fakeware_Pattern" xmlns=""><![CDATA[]]></pattern>
<pattern id="DestinationPort-Fakeware_Pattern" case-insensitive="true" xmlns=""><![CDATA[]]></pattern>
<pattern id="Protocol-Fakeware_Pattern" case-insensitive="true" xmlns=""><![CDATA[]]></pattern>
<match-group order="1" description="FWSM Test" device-type-id-override="6" xmlns="">
   <matcher field="EventName" order="1" pattern-id="EventName-Fakeware_Pattern" capture-group="1"/>
   <matcher field="SourceIp" order="1" pattern-id="SourceIp-Fakeware_Pattern" capture-group="1" />
   <matcher field="SourcePort" order="1" pattern-id="SourcePort-Fakeware_Pattern" capture-group="1"/>
   <matcher field="SourceMAC" order="1" pattern-id="SourceMAC-Fakeware_Pattern" capture-group="1" />
   <matcher field="DestinationIp" order="1" pattern-id="DestinationIp-Fakeware_Pattern" capture-group="1" />
   <matcher field="DestinationPort" order="1" pattern-id="SDestinationPort-Fakeware_Pattern" capture-group="1" />
   <matcher field="Protocol" order="1" pattern-id="Protocol-Fakeware_Pattern" capture-group="1" />
   <event-match-multiple pattern-id="EventNameId" capture-group-index="1" device-event-category="Cisco Firewall"/>
</match-group>
</device-extension>
```

Parsing basics

The preceding extension document example demonstrates some of the basic aspects of parsing:

- IP addresses
- Ports
- Protocol
- Multiple fields that use the same pattern with different groups

This example parses all FWSM events that follow the specified pattern. The fields that are parsed might not be present in those events when the events include different content.

The information that was necessary to create this configuration that was not available from the event:

- The event name is only the last 6 digits (`302015`) of the `%FWSM-session-0-302015` portion of the event.
- The FWSM has a hardcoded device event category of `Cisco Firewall`.
- The FWSM DSM uses the Cisco Pix QIDmap and therefore includes the `device-type-id-override="6"` parameter in the match group. The Pix firewall log source type ID is 6. For more informaton, see “Log Source Type IDs” on page 44).

Note: If the QID information is not specified or is unavailable, you can modify the event mapping. For more information, see the Modifying Event Mapping section in the *IBM Security QRadar SIEM Users Guide*.

Event name and device event category

An event name and a device event category are required when the QIDmap is searched. This device event category is a grouping parameter within the database that helps define like events within a device. The `event-match-multiple` at the end of the match group includes hardcoding of the category. The `event-match-multiple` uses the EventNameId pattern on the parsed event name to match up to 6 digits. This pattern is not run against the full payload, just that portion parsed as the EventName field.

The EventName pattern references the `%FWSM` portion of the events; all Cisco FWSM events contain the `%FWSM` portion. The pattern in the example matches `%FWSM` followed by any number (zero or more) of letters and dashes. This pattern match resolves the word `session` that is embedded in the middle of the event name that needs to be removed. The event severity (according to Cisco), followed by a dash and then the true event name as expected by QRadar. The `(\d{6})` string is the only string within the EventNameFWSM pattern that has a capture group.

The IP addresses and ports for the event all follow the same basic pattern: an IP address followed by a colon followed by the port number. This pattern parses two pieces of data (the IP address and the port), and specifies different capture groups in the matcher section.

```
<device-extension>
<pattern id="EventName1">(logger):</pattern>
<pattern id="DeviceTime1">time=\[(\d{2}/\w{3}/\d{4}:\d{2}:\d{2}:\d{2})\] </pattern>
<pattern id="Username">(TLSv1)</pattern>
<match-group order="1" description="Full Test">
   <matcher field="EventName" order="1" pattern-id="EventName1" capture-group="1"/>
   <matcher field="DeviceTime" order="1" pattern-id="DeviceTime1"
       capture-group="1" ext-data="dd/MMM/YYYY:hh:mm:ss"/>
   <matcher field="UserName" order="1" pattern-id="Username" capture-group="1"/>
</match-group>
</device-extension>
```

IP address and port patterns

The IP address and port patterns are four sets of one to three digits, separated by periods followed by a colon and the port number. The IP address section is in a group, as is the port number, but not the colon. The matcher sections for these fields reference the same pattern name, but a different capture group (the IP address is group 1 and the port is group 2).

The protocol is a common pattern that searches the payload for the first instance of TCP, UDP, ICMP, or GRE. The pattern is marked with the case-insensitive parameter so that any occurrence matches.

Although a second protocol pattern does not occur in the event that is used in the example, there is a second protocol pattern that is defined with an order of two. If the lowest-ordered protocol pattern does not match, the next one is attempted, and so on. The second protocol pattern also demonstrates direct substitution; there are no match groups in the pattern, but with the enable-substitutions parameter enabled, the text TCP can be used in place of protocol=6.

Creating a log source extensions document

Create log source extensions (LSX) for log sources that don't have a supported DSM, or to repair an event that has missing or incorrect information, or to parse an event when the associated DSM fails to produce a result.

For log sources that don't have an official DSM, use a Universal DSM, or UDSM, to integrate log sources. A log source extension (also known as a device extension) is then applied to the UDSM to provide the logic for parsing the logs. The LSX is based on Java regular expressions and can be used against any log protocol, such as syslog, JDBC, and LFPS. Values can be extracted from the logs and mapped to all common fields within QRadar.

When you use log source extensions to repair missing or incorrect content, any new events that are produced by the log source extensions are associated to the log source that failed to parse the original payload. Creating an extension prevents unknown or uncategorized events from being stored as unknown in IBM Security QRadar.

Follow these steps to create a log source extension:

1. Ensure that a log source is created in QRadar.

   Use Universal DSM as the log source type to handle items that are not in the list. You can also manually create a log source to prevent the logs from being automatically classified.
2. To determine what fields are available, use the **Log Activity** tab to export the logs for evaluation.
3. Use the extension document example template to determine the fields that you can use. (“Extension document template” on page 30).

   It is not necessary to use all of the fields in the template. Determine the values in the log source that can be mapped to the fields in extension document template. For more information, see “Extension document template” on page 30.
4. Remove any unused fields and their corresponding Pattern IDs from the log source extension document.
5. Upload the extension document and apply the extension to the log source.
6. Map the events to their equivalents in the QIDmap.

   This manual action on the **Log Activity** tab is used to map unknown log source events to known QRadar events so that they can be categorized and processed.

Related concepts:

“Examples of log source extensions on QRadar forum” on page 23
You can create log source extensions (LSX) for log sources that don't have a supported DSM. To help you create your own log source extensions (also known

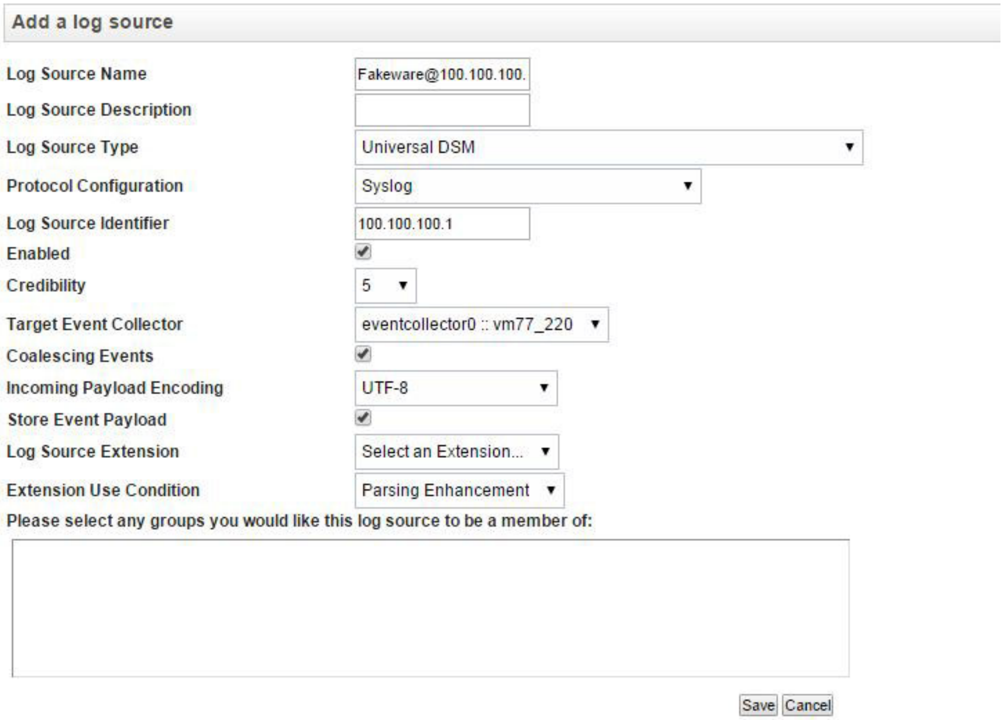
as DSM extensions), you modify existing ones that were created.

Building a Universal DSM

The first step in building a Universal DSM is to create the log source in IBM Security QRadar. When you create the log source, it prevents the logs from being automatically classified and you can export the logs for review.

Procedure

1. From the **Admin** tab, create a new source by clicking the **Log Sources** icon.
2. Click **Add**.
3. Specify the name in the **Log Source Name** field.
4. From the **Log Source Type** list, select **Universal DSM**.



Add a log source

Log Source Name	Fakeware@100.100.100.
Log Source Description	
Log Source Type	Universal DSM ▾
Protocol Configuration	Syslog ▾
Log Source Identifier	100.100.100.1
Enabled	☑
Credibility	5 ▾
Target Event Collector	eventcollector0 :: vm77_220 ▾
Coalescing Events	☑
Incoming Payload Encoding	UTF-8 ▾
Store Event Payload	☑
Log Source Extension	Select an Extension... ▾
Extension Use Condition	Parsing Enhancement ▾

Please select any groups you would like this log source to be a member of:

Figure 1. Add a log source

You might not see the **Log Source Extension** or **Extension Use Condition** unless you already applied a log source extension to the QRadar Console

5. From the **Protocol Configuration** list, specify the protocol that you want to use.

   This method is used by QRadar to get the logs from the unsupported log source.
6. For the **Log Source Identifier**, enter either the IP address or host name of the unsupported log source.
7. Click **Save** to save the new log source and close the window.
8. From the **Admin tab**, click **Deploy Changes**.

What to do next

"Exporting the logs" on page 35

Exporting the logs

Export the logs that are created after you build a Universal DSM

About this task

Typically you want a significant number of logs for review. Depending on the EPS rate of the unsupported log source, it might take several hours to obtain a comprehensive log sample.

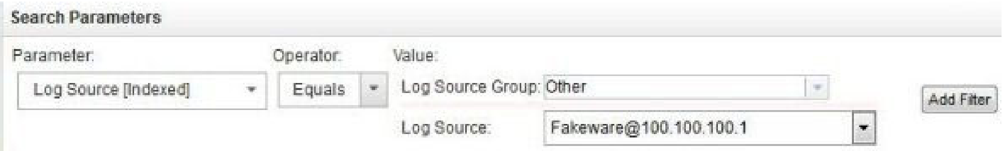
When QRadar can't detect the log source type, events are collected, but are not parsed. You can filter on these unparsed events and then review the last system notification that you received. After you reviewed the system notification, you can create a search that is based on that time frame.

Procedure

1. To look at only the events that are not parsed, filter the logs.
   a. Click the **Log Activity** tab.
   b. Click **Add Filter**.
   c. Select **Event is Unparsed**.

      Tip: Type inside the **Parameter** text box to see the **Event is Unparsed** item.
   d. Select a time frame.
   e. If you see **Information** events from system notifications, right-click to filter them out.
   f. Review the **Source IP** column to determine what device is sending the events.

      You can view the raw event payloads. Typically, manufacturers put identifiable product names in the headers, so you can set your search to **Display: Raw Events** to show the payloads without having to manually open each event. Sorting by network can also help you find a specific device where the event originated from.
2. Create a search for exporting the logs.
   a. From the **Log Activity** tab, select **Search** > **Edit Search**.
   b. For the **Time Range**, specify as enough time, for example 6 hours, from when the log source was created.
   c. Under **Search Parameters,** from the **Parameter** list, select **Log Source (Indexed)**, from the **Operator** list, select **Equals**, and from the **Log Source Group** list, select **Other**, specify the log source that was created in the when you built the Universal DSM.



Search Parameters

Parameter: Log Source [Indexed]

Operator: Equals

Value:

Log Source Group: Other

Log Source: Fakeware@100.100.100.1

   Note: Depending on your settings, you might see **Log Source** in the **Parameter** list instead of **Log Source (Indexed)**.
   d. Click **Search** to view the results.
3. Review the results in the console to check the payload.

4. Optionally, you can export the results by clicking select **Actions** > **Export to XML** > **Full Export (All Columns)**.

   Don't select **Export to CSV** because the payload might be split across multiple columns, therefore making it difficult to find the payload. XML is the preferred format for event reviews.

   a. You are prompted to download a compressed file. Open the compressed file and then open the resulting file.

   b. Review the logs.

      Event payloads are between the following tags:

      ```
      <payloadAsUTF>
      ...
      </payloadAsUTF>
      ```

      The following code shows an example payload:

      ```
      <payloadAsUTF>ecs-ep (pid 4162 4163 4164) is running... </payloadAsUTF>
      ```

      A critical step in creating a Universal DSM is reviewing the logs for usability. At a minimum, the logs must have a value that can be mapped to an event name. The event name must be a unique value that can distinguish the various log types.

      The following code shows an example of usable logs:

      ```
      May 20 17:16:14 dropbear[22331]: bad password attempt for 'root'
      from 192.168.50.80:3364
      May 20 17:16:26 dropbear[22331]: password auth succeeded for
      'root' from 192.168.50.80:3364
      May 20 16:42:19 kernel: DROP IN=vlan2 OUT=
      MAC=00:01:5c:31:39:c2:08:00 SRC=172.29.255.121
      DST=255.255.255.255 PROTO=UDP SPT=67 DPT=68
      ```

      The following example codes shows slightly less usable logs:

      ```
      Oct 26 08:12:08 loopback 1256559128 autotrace[215824]: W: trace:
      no map for prod 49420003, idf 010029a2, lal 00af0008
      Oct 26 16:35:00 sxpgbd0081 last message repeated 7 times
      Nov 24 01:30:00 sxpgbd0081 /usr/local/monitor-rrd/sxpgbd0081/.rrd
      (rc=-1, opening '/usr/local/monitor-rrd/sxpgbd0081/.rrd':
      No such file or directory)
      ```

Common regular expressions

Use regular expressions to match patterns of text in the log source file. You can scan messages for patterns of letters, numbers, or a combination of both. For example, you can create regular expressions that match source and destination IP addresses, ports, MAC addresses, and more.

The following codes shows several common regular expressions:

```
\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3} \d{1,5}
(?:[0-9a-fA-F]{2}\:){5}[0-9a-fA-F]{2} (TCP|UDP|ICMP|GRE)
\w{3}\s\d{2}\s\d{2}:\d{2}:\d{2}
\s \t .*?
```

The escape character, or "\", is used to denote a literal character. For example, "." character means "any single character" and matches A, B, 1, X, and so on. To match the "." characters, a literal match, you must use "\."

Table 33. Common regex expressions

Type	Expression
Type	\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}
IP Address	\d{1,5}

Table 33. Common regex expressions (continued)

Type	Expression
Port Number	(?:[0-9a-fA-F]{2}\:){5}[0-9a-fA-F]{2}
Protocol	(TCP|UDP|ICMP|GRE)
Device Time	\w{3}\s\d{2}\s\d{2}:\d{2}:\d{2}
Whitespace	\s
Tab	\t
Match Anything	.*?

Tip: To ensure that you don't accidentally match another characters, escape any non-digit or non-alpha character.

Building regular expression patterns

To create a Universal DSM, you use regular expressions (regex) to match strings of text from the unsupported log source.

About this task

The following example shows a log entry that is referenced in the steps.

```
May 20 17:24:59 kernel: DROP MAC=5c:31:39:c2:08:00
SRC=172.29.255.121 DST=10.43.2.10 LEN=351 TOS=0x00 PREC=0x00 TTL=64 ID=9582
PROTO=UDP SPT=67 DPT=68 LEN=331
May 20 17:24:59 kernel: PASS MAC=5c:14:ab:c4:12:59
SRC=192.168.50.10 DST=192.168.10.25 LEN=351 TOS=0x00 PREC=0x00 TTL=64
ID=9583 PROTO=TCP SPT=1057 DPT=80 LEN=331
May 20 17:24:59 kernel: REJECT
MAC=5c:ad:3c:54:11:07 SRC=10.10.10.5 DST=192.168.100.25 LEN=351
TOS=0x00 PREC=0x00 TTL=64 ID=9584 PROTO=TCP SPT=25212 DPT=6881 LEN=331
```

Procedure

1. Visually analyze the unsupported log source to identify unique patterns.

   These patterns are later translated into regular expressions.
2. Find the text strings to match.

   Tip: To provide basic error checking, include characters before and after the values to prevent similar values from being unintentionally matched. You can later isolate the actual value from the extra characters.
3. Develop pseudo-code for matching patterns and include the space character to denote the beginning and end of a pattern.

   You can ignore the quotes. In the example log entry, the event names are DROP, PASS, and REJECT. The following list shows the usable event fields.
   - EventName: " kernel: VALUE "
   - SourceMAC: " MAC=VALUE "
   - SourceIp: " SRC=VALUE "
   - DestinationIp: " DST=VALUE "
   - Protocol: " PROTO=VALUE "
   - SourcePort: " SPT=VALUE "
   - DestinationPort: " DPT=VALUE "
4. Substitute a space with the \s regular expression.

You must use an escape character for non-digit or non-alpha characters. For example, = becomes \= and : becomes \:.

5. Translate the pseduo-code to a regular expression.

Table 34. Translating pseudo-code to regular expressions

Field	Pseudo-code	Regular expression
EventName	" kernel: VALUE "	\skernel\:\s.*?\s
SourceMAC	" MAC=VALUE "	\sMAC\=(?:[0-9a-fA-F]{2}\:){5}[0-9a-fA-F]{2}\s
SourceIP	" SRC=VALUE "	\sSRC\=\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}\s
DestinationIp	" DST=VALUE "	\sDST\=\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}\s
Protocol	" PROTO=VALUE "	\sPROTO\=(TCP|UDP|ICMP|GRE)\s
SourcePort	" SPT=VALUE "	\sSPT\=\d{1,5}\s
DestinationPort	" DPT=VALUE "	\sDPT\=\d{1,5}\s

6. Specify capture groups.

   A capture group isolates a certain value in the regular expression.

   For example, in the SourcePort pattern in the previous example, you can't pass the entire value since it includes spaces and `SRC=<code>`. Instead, you specify only the port number by using a capture group. The value in the capture group is what is passed to the relevant field in IBM Security QRadar.

   Insert parenthesis around the values you that you want capture:

Table 35. Mapping regular expressions to capture groups for event fields

Field	Regular expression	Capture group
EventName	\skernel\:\s.*?\s	\skernel\:\s(.*?)\s
SourceMAC	\sMAC\=(?:[0-9a-fA-F]{2}\:){5}[0-9a-fA-F]{2}\s	\sMAC\=((?:[0-9a-fA-F]{2}\:){5}[0-9a-fA-F]{2})\s
SourceIP	\sSRC\=\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}\s	\sSRC\=(\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})\s
Destination IP	\sDST\=\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}\s	\sDST\=(\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})\s
Protocol	\sPROTO\=(TCP|UDP|ICMP|GRE)\s	\sPROTO\=((TCP|UDP|ICMP|GRE))\s
SourcePort	\sSPT\=\d{1,5}\s	\sSPT\=(\d{1,5})\s
DestinationPort	\sDPT\=\d{1,5}\s	\sDPT\=(\d{1,5})\s

7. Migrate the patterns and capture groups into the log source extensions document.

   The following code snippet shows part of the document that you use.

```
<device-extension xmlns="event_parsing/device_extension">
<pattern id="EventNameFWSM_Pattern" xmlns=""><![CDATA[%FWSM[a-zA-Z\-]*\d-(\d{1,6})]]></pattern>
<pattern id="SourceIp_Pattern" xmlns=""><![CDATA[gaddr (\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})/([\d]{1,5})]]></pattern>
<pattern id="SourceIpPreNAT_Pattern" xmlns=""><![CDATA[gaddr (\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})/([\d]{1,5})]]></pattern>
<pattern id="SourceIpPostNAT_Pattern" xmlns=""><![CDATA[laddr (\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})/([\d]{1,5})]]></pattern>
<pattern id="DestinationIp_Pattern" xmlns=""><![CDATA[faddr (\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3})/([\d]{1,5})]]></pattern>
<pattern id="Protocol_Pattern" case-insensitive="true" xmlns=""><![CDATA[(TCP|UDP|ICMP|GRE)]]></pattern>
<pattern id="Protocol_6_Pattern" case-insensitive="true" xmlns=""><![CDATA[protocol=6]]></pattern>
<pattern id="EventNameId_Pattern" xmlns=""><![CDATA[(\d{1,6})]]></pattern>
```

Uploading extension documents to QRadar

You can create multiple extension documents and then upload them and associated them to various log source types. The logic from the log source extension (LSX) is then used to parse the logs from the unsupported log source.

Extension documents can be stored anywhere before you upload to IBM Security QRadar.

Procedure

1. From the **Admin** tab, click the **Data Sources** > **Log Source Extensions**.
2. In the Add Log Source Extensions window, click **Add**.
3. Assign a name.
4. Click **Use Condition as Parsing Override**.
5. If you are using the Universal DSM, don't select the extension document as the default for a **Log Source Type**.

   By selecting the Universal DSM as the default, it affects all associated log sources. A Universal DSM can be used to define the parsing logic for multiple custom and unsupported event sources.
6. Optional: If you want to apply this log source extension to more than one instance of a log source type, select the log source type from the available **Log Source Type** list and click the add arrow to set it as the default.

   Setting the default log source type applies the log source extension to all events of a log source type, including those log sources that are automatically discovered.

   Ensure that you test the extension for the log source type first to ensure that the events are parsed correctly.
7. Click **Browse** to locate the LSX that you saved and then click **Upload**.

   QRadar validates the document against the internal XSD and verifies the validity of the document before the extension document is uploaded to the system.
8. Click **Save** and close the window.
9. Associate the log source extension to a log source.
   a. From the **Admin** tab, click **Data Sources** > **Log Sources**.
   b. Double-click the log source type that you created the extension document for.
   c. From the **Log Source Extension** list, select the document that you created.
   d. From the **Extension Use Condition** list, select **Parsing Override**.
   e. Click **Save** and close the window.

Mapping unknown events

Initially, all of the events from the Universal DSM appear as unknown in the **Log Activity** tab in QRadar. You must manually map all unknown events to their equivalents in the QID map.

Although the event names, such as DROP, DENY, and ACCEPT, might be understandable values when you see them in the log files, QRadar doesn't understand what these values represent. To QRadar, these values are strings of text that are not mapped to any known values. The values appear as expected and are treated as normalized events until you manually map them.

In some instances, such as an intrusion detection system (IDS) or an intrusion detection and prevention system (IDP) thousands of events exist and require mapping. In these situations, you can map a category as the event name instead of the itself. For example, in the following example, to reduce the number of mappings, instead of using the name field for the Event Name, use the category field instead. You can use a custom property to display the event name (Code Red v412):

```
date: "Feb 25 2010 00:43:26"; name: "SQL Slammer v312"; category: "Worm
Activity"; source ip: "100.100.200.200";⌂date:
"Feb 25 2015 00:43:26"; name: "Code Red v412"; category: "Worm Activity";
source ip: "100.100.200.200"; date: "Feb 25 2015 00:43:26"; name:
"Annoying Toolbar"; category: "Malware"; source ip: "100.100.200.200";
```

Instead of using the name field for the Event Name, use the category field instead. he actual event name, e.g. Code Red v412 can be displayed using a custom property.

Before you begin

Ensure that you uploaded the log source extension document and applied it to the Universal DSM. For more information, see “Uploading extension documents to QRadar” on page 39.

Procedure

1. From the **Log Activity** tab, click **Search** > **Edit Search**
2. From the **Time Range** options, choose enough time, such as 15 minutes, from when the log source extension was applied to the Universal DSM.
3. Under **Search Parameters**, select **Log Source [Index]** from the **Parameter** list, **Equals** from the **Operator** list and then select the log source that you created from the **Log Source Group** and the **Log Source lists**.
4. Click **Search** to view the results.

   All of the events appear as unknown.
5. Double-click an unknown entry to view the event details.
6. Click **Map Event** from the toolbar.

   The value **Log Source Event ID** displays an **EventName value**, for example, DROP, DENY, or ACCEPT, from the log source extension. The value can't be blank. A blank value indicates that there is an error in the log source extension document.
7. Map the value that is displayed as the **Log Source Event ID** to the appropriate QID.

   Use the **Browse By Category**, or **QID Search**, or both to find a value that best matches the **Log Source Event ID** value. For example, the value DROP can be mapped to the **QID Firewall Deny - Event CRE**.

   Use the QID with the Event CRE in the name. Most events are specific to a particular log source type. For example, when you map to a random firewall, **Deny QID** is similar to mapping the Universal DSM to events from another log source type. The QID entries that contain the name Event CRE are generic and are not tied to a particular log source type.
8. Repeat these steps until all unknown events are mapped successfully.

   From this point, any further events from the Universal DSM that contain that particular Log Source Event ID appear as the specified QID. Events that arrived before the QID mapping remain unknown. There is no supported method for mapping previous events to a current QID. This process must be repeated until all of the unknown event types are successfully mapped to a QID.

Parsing issues and examples

When you create a log source extension, you might encounter some parsing issues. Use these XML examples to resolving specific parsing issues.

Converting a protocol

The following example shows a typical protocol conversion that searches for TCP, UDP, ICMP, or GRE anywhere in the payload. The search pattern is surrounded by any word boundary, for example, tab, space, end of line. Also, the character case is ignored:

```
<pattern id="Protocol" case-insensitive="true" xmlns="">
<![CDATA[\b(TCP|UDP|ICMP|GRE)\b]]>
</pattern>
<matcher field="Protocol" order="1" pattern-id="Protocol" capture-group="1" />
```

Making a single substitution

The following example shows a substitution that parses the source IP address, and then overrides the result and sets the IP address to 100.100.100.100, ignoring the IP address in the payload.

This example assumes that the source IP address matches something similar to SrcAddress=10.3.111.33 followed by a comma:

```
<pattern id="SourceIp_AuthenOK" xmlns="">
<![CDATA[SrcAddress=(\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}),]]>
</pattern>

<matcher field="SourceIp" order="1" pattern-id="SourceIp_AuthenOK"
capture-group="100.100.100.100" enable-substitutions="true"/>
```

Generating a colon-separated MAC address

QRadar detects MAC addresses in a colon-separated form. Because all devices might not use this form, the following example shows how to correct that situation:

```
<pattern id="SourceMACWithDashes" xmlns="">
    <![CDATA[SourceMAC=([0-9a-fA-F]{2})-([0-9a-fA-F]{2})-([0-9a-fA-F]{2})-
    ([0-9a-fA-F]{2})-([0-9a-fA-F]{2})-([0-9a-fA-F]{2})]]>
</pattern>
 <matcher field="SourceMAC" order="1" pattern-id="
    SourceMACWithDashes" capture-group="\1:\2:\3:\4:\5:\6" />
```

In the preceding example, SourceMAC=12-34-56-78-90-AB is converted to a MAC address of 12:34:56:78:90:AB.

If the dashes are removed from the pattern, the pattern converts a MAC address and has no separators. If spaces are inserted, the pattern converts a space-separated MAC address.

Combining IP address and port

Typically an IP address and port are combined into one field, which is separated by a colon.

The following example uses multiple capture groups with one pattern:

```
pattern id="SourceIPColonPort" xmlns="">
<! [CDATA[Source=(\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}):([\d]{1,5})]]>
</pattern>

<matcher field="SourceIp" order="1" pattern-id="SourceIPColonPort" capture-group="1" />
<matcher field="SourcePort" order="1" pattern-id="SourceIPColonPort" capture-group="2" />
```

Modifying an Event Category

A device event category can be hardcoded, or the severity can be adjusted.

The following example adjusts the severity for a single event type:

```
<event-match-single event-name="TheEvent" device-event-category="Actual
Category" severity="6" send-identity="UseDSMResults" />
```

Suppressing identity change events

A DSM might unnecessarily send identity change events.

The following examples show how to suppress identity change events from being sent from a single event type and a group of events.

```
// Never send identity for the event with an EventName of Authen OK
<event-match-single event-name="Authen OK" device-event-category="ACS"
severity="6" send-identity="OverrideAndNeverSend" />

// Never send any identity for an event with an event name starting with 7,
followed by one to five other digits:
<pattern id="EventNameId" xmlns=""><![CDATA[(7\d{1,5})]]>
</pattern>

<event-match-multiple pattern-id="EventNameId" capture-group-index="1"
device-event-category="Cisco Firewall" severity="7"
send-identity="OverrideAndNeverSend"/>
```

Encoding logs

The following encoding formats are supported:

- US-ASCII
- UTF-8

You can forward logs to the system in an encoding that does not match US-ASCII or UTF-8 formats. You can configure an advanced flag to ensure that input can be re-encoded to UTF-8 for parsing and storage purposes.

For example, if you want to ensure that the source logs arrive in SHIFT-JIS (ANSI/OEM Japanese) encoding, type the following code:

```
<device-extension source-encoding=SHIFT-JIS xmlns=event_parsing/device_extension>
```

The logs are enclosed in UTF-8 format.

Formatting event dates and time stamps

A log source extension can detect several different date and time stamp formats on events.

Because device manufacturers do not conform to a standard date and time stamp format, the ext-data optional parameter is included in the log source extension to allow the DeviceTime to be reformatted. The following example shows how an event can be reformatted to correct the date and time stamp formatting:

```
<device-extension>
<pattern id="EventName1">(logger):</pattern>
<pattern id="DeviceTime1">time=\[(\d{2}/\w{3}/\d{4}:\d{2}:\d{2}:\d{2})\]</pattern>
<pattern id="Username">(TLSv1)</pattern>

<match-group order="1" description="Full Test">
   <matcher field="EventName" order="1" pattern-id="EventName1_Pattern" capture-group="1"/>
   <matcher field="DeviceTime" order="1" pattern-id="DeviceTime1_Pattern"
   capture-group="1" ext-data="dd/MMM/YYYY:hh:mm:ss"/>
   <matcher field="UserName" order="1" pattern-id="Username_Pattern" capture-group="1"/>
</match-group>
</device-extension>
```

Multiple Log Formats in a Single Log Source

Occasionally, multiple log formats are included in a single log source.

```
May 20 17:15:50 kernel: DROP IN=vlan2 OUT= MAC= SRC=67.149.62.133
DST=239.255.255.250 PROTO=UDP SPT=1900 DPT=1900
May 20 17:16:26 dropbear[22331]: password auth succeeded for 'root' from 192.168.50.80:3364
May 20 17:16:28 dropbear[22331]: exit after auth (root): Exited normally </br>
May 20 17:16:14 dropbear[22331]: bad password attempt for 'root' from 192.168.50.80:3364
```

For example, there are 2 log formats: one for firewall events, and one for authentication events. You must write multiple patterns for parsing the events. You can specify the order to be parsed. Typically, the more frequent events are parsed first, followed by the less frequent events. You can have as many patterns as required to parse all of the events. The order variable determines what order the patterns are matched in.

The following example shows multiple formats for the following fields EventName and UserName

Separate patterns are written to parse each unique log type. Both of the patterns are referenced when you assign the value to the normalized fields.

```
<pattern id="EventName-DDWRT-FW_Pattern" xmlns=""><![CDATA[kernel\:\s(.*?)\s]]></pattern>
<pattern id="EventName-DDWRT-Auth_Pattern" xmlns=""><![CDATA[sdrophear\[\d{1,5}\]|:\s(.*?\s.*?)\s]]>
</pattern>

<pattern id="UserName_DDWRT-Auth1__Pattern" xmlns=""><![CDATA[\sfor\s\'(.*?)\'s]]></pattern>
<pattern id="UserName_DDWRT-Auth2__Pattern" xmlns=""><![CDATA[\safter\sauth\s\((.*?)\)\:]]></pattern>

<match-group order="1" description="DD-WRT Device Extensions xmlns="">
   <matcher field="EventName" order="1" pattern-id="EventName-DDWRT-FW_Pattern" capture-group="1"/>
   <matcher field="EventName" order="2" pattern-id="EventName-DDWRT-Auth_Pattern" capture-group="1"/>

   <matcher field="UserName" order="1" pattern-id="UserName-DDWRT-Auth1_Pattern" capture-group="1"/>
   <matcher field="UserName" order="2" pattern-id="UserName-DDWRT-Auth2_Pattern" capture-group="1"/>
```

Parsing a CSV log format

A CSV-formatted log file can use a single parser that has multiple capture groups. It is not always necessary to create multiple Pattern IDs when you parse this log type.

About this task

The following log sample is used:

```
Event,User,Source IP,Source Port,Destination IP,Destination Port
Failed Login,bjones,192.168.50.100,1024,10.100.24.25,22
Successful Login,nlabadie,192.168.64.76,1743,10.100.24.25,110
Privilege Escalation,bjones,192.168.50.100,1028,10.100.1.100,23
```

Procedure

1. Create a parser that matches all relevant values by using the previous patterns.

   ```
   .*?\,.*?\,\d{1,3}\.\d{1,3}\.\d{1,3}\.\d{1,3}
   \,\d{1,5}\,\d{1,3}\.\d{1,3} \.\d{1,3}\.\d{1,3}\,\d{1,5}
   ```

2. Place the capture groups around each value:

   ```
   (.*?)\,(.*?)\,(\d{1,3}\.\d{1,3}\.\d{1,3}\.
   \d{1,3})\,(\d{1,5})\,(\d{1,3} \.\d{1,3}\.\d{1,3}\.\d{1,3})\,(\d{1,5})
   ```

3. Map the field that each capture group is mapped to, incrementing the value as you move.

   ```
   1 = Event, 2 = User, 3 = Source IP,
   4 = Source Port, 5 = Destination IP, 6 = Destination Port
   ```

4. Include the values in the log source extension by mapping the capture group to the relevant event.

   The following code shows a partial example of mapping the capture group to the relevant event.

   ```
   <pattern id="CSV-Parser_Pattern" xmlns=""><![CDATA 9.*?)\,(.*?)\,(\d{1,3}\.\{1,3}\.{1,3}]]></pattern>
   <match-group order="1" description="Log Source Extension xmlns="">
      <matcher field="EventName" order="1" pattern-id="CSV-Parser_Pattern" capture-group="1"/>
      <matcher field="SourceIP" order="1" pattern-id="CSV-Parser_Pattern" capture-group="3"/>
      <matcher field="SourcePort" order="1" pattern-id="CSV-Parser_Pattern" capture-group="4"/>
      <matcher field="DestinationIP" order="1" pattern-id="CSV-Parser_Pattern" capture-group="5"/>
      <matcher field="DestinationPort" order="1" pattern-id="CSV-Parser_Pattern" capture-group="6"/>
      <matcher field="UserName" order="1" pattern-id="CSV-Parser_Pattern" capture-group="2"/>
   ```

5. Upload the log source extension.
6. Map the events.

Related tasks:

"Mapping unknown events" on page 39
Initially, all of the events from the Universal DSM appear as unknown in the **Log Activity** tab in QRadar. You must manually map all unknown events to their equivalents in the QID map.

Log Source Type IDs

IBM Security QRadar supports a number of log sources and each log source has an identifier. Use the Log Source Type IDs in a `match-group` statement:

The following table lists the supported log source type and their IDs.

Table 36. Log Source Type ID

ID	Log Source Type
2	Snort Open Source IDS
3	Check Point Firewall-1
4	Configurable Firewall Filter
5	Juniper Networks Firewall and VPN
6	Cisco PIX Firewall
7	Configurable Authentication message filter
9	Enterasys Dragon Network IPS
10	Apache HTTP Server
11	Linux OS
12	Microsoft Windows Security Event Log
13	Windows IIS
14	Linux iptables Firewall

Table 36. Log Source Type ID (continued)

ID	Log Source Type
15	IBM Proventia Network Intrusion Prevention System (IPS)
17	Juniper Networks Intrusion Detection and Prevention (IDP)
19	TippingPoint Intrusion Prevention System (IPS)
20	Cisco IOS
21	Nortel Contivity VPN Switch
22	Nortel Multiprotocol Router
23	Cisco VPN 3000 Series Cntrator
24	Solaris Operating System Authentication Messages
25	McAfee IntruShield Network IPS Appliance
26	Cisco CSA
28	Enterasys Matrix E1 Switch
29	Solaris Operating System Sendmail Logs
30	Cisco Intrusion Prevention System (IDS)
31	Cisco Firewall Services Module (FWSM)
33	IBM Proventia Management SiteProtector
35	Cyberguard FW/VPN KS Family
36	Juniper Networks Secure Access (SA) SSL VPN
37	Nortel Contivity VPN Switch
38	Top Layer Intrusion Prevention System (IPS)
39	Universal DSM
40	Tripwire Enterprise
41	Cisco Adaptive Security Appliance (ASA)
42	Niksun 2005 v3.5
45	Juniper Networks Network and Security Manager (NSM)
46	Squid Web Proxy
47	Ambiron TrustWave ipAngel Intrusion Prevention System (IPS)
48	Oracle RDBMS Audit Records
49	F5 Networks BIG-IP LTM
50	Solaris Operating System DHCP Logs
55	Array Networks SSL VPN Access Gateway
56	Cisco CatOS for Catalyst Switches
57	ProFTPD Server
58	Linux DHCP Server
59	Juniper Networks Infranet Controller
64	Juniper JunOS Platform

Table 36. Log Source Type ID (continued)

ID	Log Source Type
68	Enterasys Matrix K/N/S Series Switch
70	Extreme Networks ExtremeWare Operating System (OS)
71	Sidewinder G2 Security Appliance
73	Fortinet FortiGate Security Gateway
78	SonicWall UTM/Firewall/VPN device
79	Vericept Content 360
82	Symantec Gateway Security (SGS) Appliance
83	Juniper Steel Belted Radius
85	IBM AIX Server
86	Metainfo MetaIP
87	SymantecSystemCenter
90	Cisco ACS
92	Forescout CounterACT
93	McAfee ePolicy Orchestrator
95	CiscoNAC Appliance
96	TippingPoint X Series Appliances
97	Microsoft DHCP Server
98	Microsoft IAS Server
99	Microsoft Exchange Server
100	Trend Interscan VirusWall
101	Microsoft SQL Server
102	MAC OS X
103	Bluecoat SG Appliance
104	Nortel Switched Firewall 6000
106	3Com 8800 Series Switch
107	Nortel VPN Gateway
108	Nortel Threat Protection System (TPS) Intrusion Sensor
110	Nortel Application Switch
111	Juniper DX Application Acceleration Platform
112	SNARE Reflector Server
113	Cisco 12000 Series Routers
114	Cisco 6500 Series Switches
115	Cisco 7600 Series Routers
116	Cisco Carrier Routing System
117	Cisco Integrated Services Router
118	Juniper M-Series Multiservice Edge Routing
120	Nortel Switched Firewall 5100
122	Juniper MX-Series Ethernet Services Router

Table 36. Log Source Type ID (continued)

ID	Log Source Type
123	Juniper T-Series Core Platform
134	Nortel Ethernet Routing Switch 8300/8600
135	Nortel Ethernet Routing Switch 2500/4500/5500
136	Nortel Secure Router
138	OpenBSD OS
139	Juniper Ex-Series Ethernet Switch
140	Sysmark Power Broker
141	Oracle Database Listener
142	Samhain HIDS
143	Bridgewater Systems AAA Service Controller
144	Name Value Pair
145	Nortel Secure Network Access Switch (SNAS)
146	Starent Networks Home Agent (HA)
148	IBM AS/400 iSeries
149	Foundry Fastiron
150	Juniper SRX Series Services Gateway
153	CRYPTOCard CRYPTOShield
154	Imperva Securesphere
155	Aruba Mobility Controller
156	Enterasys NetsightASM
157	Enterasys HiGuard
158	Motorola SymbolAP
159	Enterasys HiPath
160	Symantec Endpoint Protection
161	IBM RACF
163	RSA Authentication Manager
164	Redback ASE
165	Trend Micro Office Scan
166	Enterasys XSR Security Routers
167	Enterasys Stackable and Standalone Switches
168	Juniper Networks AVT
169	OS Services Qidmap
170	Enterasys A-Series
171	Enterasys B2-Series
172	Enterasys B3-Series
173	Enterasys C2-Series
174	Enterasys C3-Series
175	Enterasys D-Series

Table 36. Log Source Type ID (continued)

ID	Log Source Type
176	Enterasys G-Series
177	Enterasys I-Series
178	Trend Micro Control Manager
179	Cisco IronPort
180	Hewlett Packard UniX
182	Cisco Aironet
183	Cisco Wireless Services Module (WiSM)
185	ISC BIND
186	IBM Lotus Domino
187	HP Tandem
188	Sentrigo Hedgehog
189	Sybase ASE
191	Microsoft ISA
192	Juniper SRC
193	Radware DefensePro
194	Cisco ACE Firewall
195	IBM DB2
196	Oracle Audit Vault
197	Sourcefire Defense Center
198	Websense V Series
199	Oracle RDBMS OS Audit Record
206	Palo Alto PA Series
208	HP ProCurve
209	Microsoft Operations Manager
210	EMC VMWare
211	IBM WebSphere Application Server
213	F5 Networks BIG-IP ASM
214	FireEye
215	Fair Warning
216	IBM Informix
217	CA Top Secret
218	Enterasys NAC
219	System Center Operations Manager
220	McAfee Web Gateway
221	CA Access Control Facility (ACF2)
222	McAfee Application / Change Control
223	Lieberman Random Password Manager
224	Sophos Enterprise Console
225	NetApp Data ONTAP
226	Sophos PureMessage

Table 36. Log Source Type ID (continued)

ID	Log Source Type
227	Cyber-Ark Vault
228	Itron Smart Meter
230	Bit9 Parity
231	IBM IMS
232	F5 Networks FirePass
233	Citrix NetScaler
234	F5 Networks BIG-IP APM
235	Juniper Networks vGW
239	Oracle BEA WebLogic
240	Sophos Web Security Appliance
241	Sophos Astaro Security Gateway
243	Infoblox NIOS
244	Tropos Control
245	Novell eDirectory
249	IBM Guardium
251	Stonesoft Management Center
252	SolarWinds Orion
254	Great Bay Beacon
255	Damballa Failsafe
258	CA SiteMinder
259	IBM z/OS
260	Microsoft SharePoint
261	iT-CUBE agileSI
263	Digital China Networks DCS and DCRS Series switch
264	Juniper Security Binary Log Collector
265	Trend Micro Deep Discovery
266	Tivoli Access Manager for e-business
268	Verdasys Digital Guardian
269	Hauwei S Series Switch
271	HBGary Active Defense
272	APC UPS
272	Cisco Wireless LAN Controller
276	IBM Customer Information Control System (CICS)
278	Barracuda Spam & Virus Firewall
279	Open LDAP
280	Application Security DbProtect
281	Barracuda Web Application Firewall
283	Huawei AR Series Router

Table 36. Log Source Type ID (continued)

ID	Log Source Type
286	IBM AIX Audit
289	IBM Tivoli Endpoint Manager
290	Juniper Junos WebApp Secure
291	Nominum Vantio
292	Enterasys 800-Series Switch
293	IBM zSecure Alert
294	IBM Security Network Protection (XGS)
295	IBM Security Identity Manager
296	F5 Networks BIG-IP AFM
297	IBM Security Network IPS (GX)
298	Fidelis XPS
299	Arpeggio SIFT-IT
300	Barracuda Web Filter
302	Brocade FabricOS
303	ThreatGRID Malware Threat Intelligence Platform
304	IBM Security Access Manager for Enterprise Single Sign-On
306	Venustech Venusense Unified Threat Management
307	Venustech Venusense Firewall
308	Venustech Venusense Network Intrusion Prevention System
309	ObserveIT
311	Pirean Access: One
312	Venustech Venusense Security Platform
313	PostFix MailTransferAgent
314	Oracle Fine Grained Auditing
315	VMware vCenter
316	Cisco Identity Services Engine
318	Honeycomb Lexicon File Integrity Monitor
319	Oracle Acme Packet SBC
320	Juniper WirelessLAN
330	Arbor Networks Peakflow SP
331	Zscaler Nss
332	Proofpoint Enterprise Protection/Enterprise Privacy
338	Microsoft Hyper-V
339	Cilasoft QJRN/400
340	Vormetric Data Security
341	SafeNet DataSecure/KeySecure

Table 36. Log Source Type ID (continued)

ID	Log Source Type
343	STEALTHbits StealthINTERCEPT
344	Juniper DDoS Secure
345	Arbor Networks Pravail
346	Trusteer Apex
348	IBM Security Directory Server
349	Enterasys A4-Series
350	Enterasys B5-Series
351	Enterasys C5-Series
354	Avaya VPN Gateway
356	DG Technology MEAS
358	CloudPassage Halo
359	CorreLog Agent for IBM zOS
360	WatchGuard Fireware OS
361	IBM Fiberlink MaaS360
362	Trend Micro Deep Discovery Analyzer
363	AccessData InSight
364	BM Privileged Session Recorder
367	Universal CEF
369	FreeRADIUS
370	Riverbed SteelCentral NetProfiler
372	SSH CryptoAuditor
373	IBM WebSphere DataPower
374	Symantec Critical System Protection
375	Kisco Information Systems SafeNet/i
376	IBM Federated Directory Server
378	Lastline Enterprise
379	genua genugate
383	Oracle Enterprise Manager

Chapter 3. Log source extension management

You can create log source extensions to extend or modify the parsing routines of specific devices.

A *log source extension* is an XML file that includes all of the regular expression patterns that are required to identify and categorize events from the event payload. Extension files can be used to parse events when you must correct a parsing issue or you must override the default parsing for an event from a DSM. When a DSM does not exist to parse events for an appliance or security device in your network, an extension can provide event support. The **Log Activity** tab identifies log source events in these basic types:

- Log sources that properly parse the event. Properly parsed events are assigned to the correct log source type and category. In this case, no intervention or extension is required.
- Log sources that parse events, but have a value **Unknown** in the **Log Source** parameter. Unknown events are log source events where the log source type is identified, but the payload information cannot be understood by the DSM. The system cannot determine an event identifier from the available information to properly categorize the event. In this case, the event can be mapped to a category or a log source extension can be written to repair the event parsing for unknown events.
- Log sources that cannot identify the log source type and have a value of **Stored** event in the **Log Source** parameter. Stored events require you to update your DSM files or write a log source extension to properly parse the event. After the event parses, you can then map the events.

Before you can add a log source extension, you must create the extension document. The extension document is an XML document that you can create with any common word processing or text editing application. Multiple extension documents can be created, uploaded, and associated with various log source types. The format of the extension document must conform to a standard XML schema document (XSD). To develop an extension document, knowledge of and experience with XML coding is required.

Adding a log source extension

You can add a log source extension to extend or modify the parsing routines of specific devices.

Procedure

1. Click the **Admin** tab.
2. Click the **Log Source Extensions** icon.
3. Click **Add**.
4. From the **Use Condition** list, select one of the following options:

Option	Description
Parsing Enhancement	Select this option when the device support module (DSM) correctly parses most fields for the log source. The incorrectly parsed field values are enhanced with the new XML values.
Parsing Override	Select this option when the device support module (DSM) is unable to parse correctly. The log source extension completely overrides the failed parsing by the DSM and substitutes the parsing with the new XML values.

5. From the **Log Source Types** list, select one of the following options:

Option	Description
Available	Select this option when the device support module (DSM) correctly parses most fields for the log source. The incorrectly parsed field values are enhanced with the new XML values.
Set to default for	Select log sources to add or remove from the extension parsing. You can add or remove extensions from a log source. When a log source extension is **Set to default for** a log source, new log sources of the same **Log Source Type** use the assigned log source extension.

6. Click **Browse** to locate your log source extension XML document.
7. Click **Upload**. The contents of the log source extension is displayed to ensure that the proper extension file is uploaded. The extension file is evaluated against the XSD for errors when the file is uploaded.
8. Click **Save**.

Results

If the extension file does not contain any errors, the new log source extension is created and enabled. It is possible to upload a log source extension without applying the extension to a log source. Any change to the status of an extension is applied immediately and managed hosts or Consoles enforce the new event parsing parameters in the log source extension.

What to do next

On the **Log Activity** tab, verify that the parsing patterns for events is applied correctly. If the log source categorizes events as **Stored**, the parsing pattern in the log source extension requires adjustment. You can review the extension file against log source events to locate any event parsing issues.

Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program, or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any non-IBM product, program, or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not grant you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
North Castle Drive
Armonk, NY 10504-1785 U.S.A.

For license inquiries regarding double-byte character set (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

Intellectual Property Licensing
Legal and Intellectual Property Law
IBM Japan Ltd.
19-21, Nihonbashi-Hakozakicho, Chuo-ku
Tokyo 103-8510, Japan

The following paragraph does not apply to the United Kingdom or any other country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS PUBLICATION "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions of the publication. IBM may make improvements and/or changes in the product(s) and/or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
170 Tracer Lane,
Waltham MA 02451, USA

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement, IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is no guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

All IBM prices shown are IBM's suggested retail prices, are current and are subject to change without notice. Dealer prices may vary.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information softcopy, the photographs and color illustrations may not appear.

Trademarks

IBM, the IBM logo, and ibm.com® are trademarks or registered trademarks of International Business Machines Corporation in the United States, other countries, or both. If these and other IBM trademarked terms are marked on their first occurrence in this information with a trademark symbol (® or ™), these symbols indicate U.S. registered or common law trademarks owned by IBM at the time this information was published. Such trademarks may also be registered or common law trademarks in other countries. A current list of IBM trademarks is available on the Web at Copyright and trademark information (www.ibm.com/legal/copytrade.shtml).

The following terms are trademarks or registered trademarks of other companies:

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Oracle and/or its affiliates.

Linux is a registered trademark of Linus Torvalds in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names may be trademarks or service marks of others.

Privacy policy considerations

IBM Software products, including software as a service solutions, (“Software Offerings”) may use cookies or other technologies to collect product usage information, to help improve the end user experience, to tailor interactions with the end user or for other purposes. In many cases no personally identifiable information is collected by the Software Offerings. Some of our Software Offerings can help enable you to collect personally identifiable information. If this Software Offering uses cookies to collect personally identifiable information, specific information about this offering’s use of cookies is set forth below.

Depending upon the configurations deployed, this Software Offering may use session cookies that collect each user’s session id for purposes of session management and authentication. These cookies can be disabled, but disabling them will also eliminate the functionality they enable.

If the configurations deployed for this Software Offering provide you as customer the ability to collect personally identifiable information from end users via cookies and other technologies, you should seek your own legal advice about any laws applicable to such data collection, including any requirements for notice and consent.

For more information about the use of various technologies, including cookies, for these purposes, See IBM’s Privacy Policy at http://www.ibm.com/privacy and IBM’s Online Privacy Statement at http://www.ibm.com/privacy/details the section entitled “Cookies, Web Beacons and Other Technologies” and the “IBM Software Products and Software-as-a-Service Privacy Statement” at http://www.ibm.com/software/info/product-privacy.

Index

Printed in USA